জিএলটি ইশকুল পাঠমালা

গ্নু-লিনাক্স : একটি ব্যক্তিগত যাত্রা

ত্রিদিব সেনগুপ্ত

ফ্রি সফটওয়ার ফাউন্ডেশন ইন্ডিয়া

মুদ্রণ সংক্রান্ত খুঁটিনাটি
মুদ্রণ সংক্রান্ত খুঁটিনাটি
মুদ্রণ সংক্রান্ত খুঁটিনাটি
মুদ্রণ সংক্রান্ত খুঁটিনাটি
মুদ্রণ সংক্রান্ত খুঁটিনাটি
মুদ্রণ সংক্রান্ত খুঁটিনাটি
মুদ্রণ সংক্রান্ত খুঁটিনাটি
মুদ্রণ সংক্রান্ত খুঁটিনাটি

গ্রন্থস্বত্ব (c) : দীপঙ্কর দাশ, চন্দনগড়, মধ্যমগ্রাম, ২৪ পরগনা (উত্তর), পশ্চিমবঙ্গ, ভারত, পিন ৭০০১২৯, ইমেল ddipankardas@hotmail.com, ফোন ২৫৩৮৫৬৩৮। লেখককে জানিয়ে, মুদ্রিত বা বৈদ্যুতিন যে কোনো মাধ্যমেই, এই পাঠমালার যে কোনো অংশের বা সমগ্রের অবিকল প্রতিলিপি এবং পুনর্বন্টণ করা চলবে, যদি এই গ্রন্থস্বত্ব এবং অনুমতি সংযুক্ত থাকে।

মূল্য সংক্রান্ত খুঁটিনাটি
মূল্য সংক্রান্ত খুঁটিনাটি

বিক্রির অর্থ ব্যয় হবে ফ্রি সফটওয়ার সংক্রান্ত কাজে, গ্নু-লিনাক্স আন্দোলনের কেউ এই বই থেকে কোনো আয় করছে না।

মহম্মদকে,

যদি কখনো পড়িস

গ্নু-লিনাক্স ইশকুল

জিএলটি মধ্যমগ্রাম
glt-mad@ilug-cal.org

সচরাচর ভূমিকা এবং সূচীপত্র বলে যা থাকে তাকে মেটাডেটা বলে ডাকার প্ল্যানটা এসেছিল এই পাঠমালার আট নম্বর দিনটা লিখতে লিখতে। ফাইলসিস্টেমের মেটাডেটার এর চেয়ে ভালো উদাহরণ আর হয়না। তথ্যের বিষয়ে তথ্য। এই পাঠমালায় যে তথ্য আছে তার বিষয়ে তথ্যটা এখানে রইল।

।। মেটাপাঠমালা।।

১।। কৃতজ্ঞতা কাম ভূমিকা

পাঠমালার মধ্যে বারবার এসেছে এফএসএফ-এর, কোলকাতা লাগ-এর, জিএলটি-র নানা জনের কথা। খুব এসেছে অরিজিত (মজুমদার), অশেষ (সেনগুপ্ত), ইন্দ্রনীল (দাশগুপ্ত), তথাগত (বন্দোপাধ্যায়), সঙ্কর্ষণ (মুখোপাধ্যায়), সায়মিন্দু (দাশগুপ্ত) এরা সবাই। আরো অনেকে। পাঠমালাটা নিয়ে ওদের আগ্রহ আর আবেগ আমার চেয়ে কোনো অংশেই কম ছিলনা, নভেম্বর ২০০৩ থেকে জুন ২০০৪ এই আট মাসে। কিন্তু, এর জন্যে আমার দিক থেকে কোনো কৃতজ্ঞতার কোনো মানেই হয়না, কারণ ওরা কেউই এটা আমার জন্যে করেনি, করেছে গ্নু-লিনাক্সের জন্যে। ওদের কৃতজ্ঞতা জানাতে হলে, নিজেকে সবচেয়ে বেশি জানাতে হয়, সবার মধ্যে সবচেয়ে বেশি খেটেছি আমি।

কৃতজ্ঞতা যদি জানাতেই হয়, জানানো উচিত তাদের যারা গ্নু-লিনাক্সের কেউ না-হয়েও পাঠমালাটার সঙ্গে জড়িয়ে গেছে। আমার প্রতিবেশী বাবলুদার মেয়ে প্রিয়াঙ্কা, ওর ক্লাস সিক্সের জ্যামিতি বাক্স থেকে সেট স্কোয়ার নিয়ে কনকনে ঠান্ডা রাত্তিরে বসে ছয় সাত আটের ছবিগুলো আঁকা। কিন্তু ওরও লাভ হয়েছে এতে, সেই কদিন জ্যামিতি কষতে হয়নি। কিম্বা মার্জিনাল কৃতজ্ঞতার পাত্র হতে পারে আমার ছাত্র-কাম-বন্ধু জয়দীপ (দত্ত), দুটো এলিট সেট স্কোয়ার আমায় উপহার দেওয়ায়, তবে তদ্দিনে প্রথম খসড়ার কাজ প্রায় শেষ। কিছুটা কৃতজ্ঞতার পাত্র আমার বন্ধু জয়ন্তী (বসু), লিখে চলাকালীন পাঠমালাটা নিয়ে ওর জ্যান্ত প্রশংসার জন্যে, দিনে ষোল সতেরো আঠেরো ঘন্টা করে লিখে চলতে গেলে ওসব একটু লাগেই। তবে পাঠমালা চলতে চলতে অজস্র টুকরোয় আমার পাঠানো পিডিএফে ওর ইমেল অ্যাকাউন্ট জ্যাম করে দেওয়ার দোষে এত গালাগাল করেছে, তাতে শোধ হয়ে গেছে। আর মানুকে কৃতজ্ঞতা জানাব কী, এবং আলাদা করে একটা বইয়ের জন্যে? নিজের বৌকে বরকে গুডমর্নিং গুডনাইট জানানোর মত মার্কিন এখনো হয়ে পারিনি। তাহলে, কাকেই বা জানাব? অশোকদা (দে) শুধু এই আটমাসে না, যখনই যা ডাউনলোড যা সফটওয়ার লেগেছে সেগুলো খুব ভালোবেসে যোগাড় করে দেয়, কিন্তু অশোকদা দেবাশিষ (দাস) পিউ (ভদ্র) এরা সবাই জিএলটির, মানে গ্নু-লিনাক্সের। তাহলে?

ও, একজন পাওয়া গেছে। পড়লেই জানতে পারবেন, আসলে পাঠমালাটা লেখা হয়েই উঠল আমার পা-ভাঙায়। তখন ডাক্তারবাবু সুস্মিত রায় ভারি সাহায্য করেছিলেন। ভালো ডাক্তার, ডাক্তার কেন, লোকই, এত কম হয়। ওকে জানানোই যায় কৃতজ্ঞতা। পাঠমালাটা ছাপা হলে ওকে এককপি দেব, কোনো গালাগাল খাওয়ারও চান্স থাকবেনা, গ্নু-লিনাক্সে সক্রিয় উৎসাহ ছাড়া এটা পড়ে ওঠা শক্ত, খুবই শক্ত।

২।। সূচীপত্র

গোটা পাঠমালাটা শূন্য থেকে দশ এই এগারোটা দিনে ভাগ করা। এগারোটা দিনে মোট শব্দ আছে ১৮৮,৩৭২। এই এগারোটা দিনে মোট পাতার সংখ্যা ৩৭৪, এফোর সাইজে। অনেকগুলো দিনেই সেকশন নাম্বারিং এমনই উচ্চস্তরের যে, খারাপ সেকশন নাম্বারিং-এর দৃষ্টান্ত হতে পারে পাঠমালাটা। কিন্তু এত এত ক্রস-যোগাযোগ আছে, সেই সমস্তগুলো ধরে ধরে বদলাতে হবে, আর পাল্টাতে গেলাম না।

।। দিন শূন্য।। * ০০১—০৩৪

০.০।। সামনে একটা আস্ত পিসি * ০০১
০.১।। কাজের ভিত্তিতে একটা কম্পিউটারের গঠন * ০০৩
০.২।। ইনপুট বা কম্পিউটারে তথ্য ঢোকা * ০০৫
০.৩।। আউটপুট বা কম্পিউটার থেকে তথ্য-বেরোনো * ০১০
০.৪।। ইনফর্মেশন প্রসেসিং বা তথ্য চটকানো * ০১৩
০.৫।। ইনফর্মেশন হোল্ডিং বা তথ্য ধরে রাখা * ০১৪
০.৫.১।। অনুদ্বায়ী বা স্থায়ী স্মৃতি — রম * ০১৬
০.৫.২।। উদ্বায়ী স্মৃতি — র‍্যাম * ০১৬
০.৫.৩।। তথ্য রাখার ভাঁড়ার * ০১৮
০.৬।। তথ্য নিয়ন্ত্রণ বা কন্ট্রোল ডিভাইস * ০২০
০.৭।। একটা সারণী — কাজ ভার্সাস উপাদান * ০২১
০.৮।। কম্পিউটার এবং তথ্য * ০২৩
০.৮.১।। তথ্যের পরিমাপ * ০২৪
০.৮.২।। আঙ্কিক তথ্যের আকার ও পরিমাণ * ০২৪
০.৮.২.১।। এইরে, অঙ্ক কষাচ্ছে * ০২৫
০.৮.২. ২।। ফেরত এলাম ম্যাজিকে * ০২৬
০.৮.২.৩।। আঙ্কিক তথ্যের বাইনারি আকার * ০২৭
০.৮.৩।। একটা টেক্সট ফাইলে ভরে রাখা তথ্যের পরিমাপ * ০২৮
০.৯।। বিট বাইট ওয়ার্ড — স্মৃতির একক * ০৩১

।। দিন এক।। * ০৩৫—০৬০

১।। কাজের কেন্দ্রীয় সংস্থা — সিপিইউ এবং মেমরি * ০৩৫
১.১।। সিপিইউ এবং তার রেজিস্টার-কাঠামো * ০৩৭
১.২।। মেমরি এবং তথ্য * ০৪০
২।। অ্যাপ্লিকেশন, সফটওয়ার, অপারেটিং সিস্টেম — সবই প্রোগ্রাম * ০৪৪
২. ১।। ভৌত উপাদান — কম্পিউটার সিস্টেমের ভিত * ০৪৬
২.২।। মাইক্রোআর্কিটেকচার আর মেশিন ল্যাংগুয়েজ — হার্ডওয়ার থেকে সফটওয়ার * ০৪৯
৩।। অপারেটিং সিস্টেম * ০৫০
৪।। সিস্টেম প্রোগ্রাম এবং কমান্ড ইন্টারপ্রিটার * ০৫২
৫।। এডিটর * ০৫৪
৬।। কম্পাইলার * ০৫৬
৭।। অ্যাপ্লিকেশনস * ০৫৯

।। দিন দুই ।। * ০৬১—০৮০

১।। দুটো কাজ : করছে এক জন * ০৬১
২।। মেশিন মেশিনতর মেশিনোত্তর — অপারেটিং সিস্টেম মানে পরিবর্ধিত মেশিন * ০৬৩
৩।। ভৌত উপাদানদের নিয়ে তৈরি সাম্রাজ্য * ০৬৭
৪।। অপারেটিং সিস্টেম মানে রসদের ভাঁড়ারঘরের দায়িত্ব * ০৬৯
৫।। সমান্তরাল বহুক্রিয়া বা মাল্টিপ্লেক্সিং * ০৭১
৬।। আইও ডিভাইস, ডিভাইস কন্ট্রোলার এবং ইন্টারাপ্ট * ০৭৩
৭।। আইও তথ্য, ইন্টেরাপ্ট , ডিএমএ * ০৭৬
৭.১।। ব্যস্ত-অপেক্ষা বা বিজি-ওয়েটিং * ০৭৭
৭.২।। ইন্টারাপ্ট বা ব্যাঘাত পদ্ধতি * ০৭৭
৭.৩।। ডিএমএ বা সরাসরি-স্মৃতি-সংযোগ পদ্ধতি * ০৭৯
৮।। বাস-যোগাযোগ * ০৭৯

।। দিন তিন ।। * ০৮১—১২০

১।। কম্পিউটার হার্ডওয়ারের প্রাগ-ইতিহাস * ০৮১
১.১।। অ্যাবাকাস * ০৮৪
১.২।। লগারিদম ও স্লাইড রুল * ০৮৬
১.৩।। স্লাইড রুল থেকে ব্যাবেজ * ০৮৯
১.৪।। চার্লস ব্যাবেজ এবং অঙ্কের ইঞ্জিন * ০৯২
২।। কম্পিউটার চিন্তনের বদলের ইতিহাস * ১০১
২.১।। প্রথম প্রোগ্রামার আডা লাভলেস এবং রগচটা প্রতিভা চার্লস ব্যাবেজ * ১০১
২.২।। জর্জ বুলি — চিন্তার ব্যাকরণ * ১০৮
২.৩।। বুলির লজিক আর শ্যাননের সার্কিট * ১১২

।। দিন চার ।। * ১২১—১৫০

১।। শুরুর কয়েকজন মানুষ এবং মেশিন * ১২১
১.১।। আইকেন, হপার, মার্ক ওয়ান * ১২২
১.২।। জন ভন নয়মান এবং ফলিত গণিত * ১২৩
১.৩।। মশলে, একার্ট, এনিয়াক * ১২৫
১.৪।। কনরাড জুসে, জেড সিরিজ, প্লানকালকুল * ১২৬
২।। প্রথম জেনারেশন কম্পিউটার — ১৯৪৫-৫৫ * ১২৮
৩।। দ্বিতীয় জেনারেশন কম্পিউটার — ১৯৫৫-৬৫ * ১২৯
৪।। ফোর্ট্রান মনিটরিং সিস্টেম — অপারেটিং সিস্টেমের আদিপুরুষ * ১৩২
৫।। তৃতীয় জেনারেশন কম্পিউটার — ১৯৬৫-১৯৮০ * ১৩৩
৬।। মাল্টিপ্রোগ্রামিং * ১৩৫
৭।। সময়ভাগের ব্যবস্থা — সিটিএসএস * ১৩৬
৮।। বহু নদীর মোহনা — মাল্টিক্স * ১৩৭
৯।। মিনিকম্পিউটার, ইউনিক্স, সি * ১৩৮
১০।। ইউনিক্স, মিনিক্স, লিনাক্স * ১৪১
১১।। চতুর্থ জেনারেশন কম্পিউটার — ১৯৮০ থেকে চলছে ... * ১৪৪

১২।। গ্রাফিকাল ইউজার ইন্টারফেস এবং নেটওয়ার্কিং * ১৪৫
১৩।। একই সময়ের শরীরে বিভিন্ন সময় * ১৪৮

।। দিন পাঁচ।। * ১৫১—১৯০

১।। ইউনিক্স — মেশিননিরপেক্ষ অপারেটিং সিস্টেম * ১৫১
২।। ইউনিক্স থেকে গ্নু-লিনাক্স * ১৫৪
৩।। ইউনিক্স থেকে গ্নু-লিনাক্স — সালগত ক্রমপঞ্জী * ১৫৭
৪।। কারনেল আর শেল — অপারেটিং সিস্টেমের শাঁস আর খোসা * ১৬৩
৫।। ভৌত উপাদানকে ভুলবেন না * ১৬৮
৬।। কম্পিউটারে প্রাণ-প্রতিষ্ঠা — বুটলোডার * ১৬৯
৭।। কারনেল নিজের জমি বুঝে নিচ্ছে — ইনিট * ১৭৪
৮।। তুমি কে গা? — লগ-ইন * ১৭৮
৯।। লগ-ইনকে একটু খতিয়ে দেখা — পাসওয়ার্ড * ১৮১
১০।। সিস্টেম মানে জাগ্রত এবং ঘুমন্ত প্রসেসের চিড়িয়াখানা * ১৮৫
১১।। আর একটু খোসা চর্চা * ১৮৭

।। দিন ছয়।। * ১৯১—২২২

১।। কয়েকপিস কমান্ড * ১৯১
২ ।। ফাইল * ১৯৮
২.১।। ফাইলনাম এবং এক্সটেনশন * ১৯৯
২.২।। ফাইলের গঠন * ২০২
২.৩।। রেগুলার ফাইল * ২০৩
২.৪।। ডিরেক্টরি ফাইল * ২০৫
২.৫।। ডিভাইস ফাইল * ২০৬
২.৬।। এক খাবলা পুনশ্চ * ২০৭
৩ ।। এক স্তরের সরলতম ডিরেক্টরি সিস্টেম * ২০৮
৩.১।। দুই স্তরের ডিরেক্টরি কাঠামো * ২০৯
৩.২।। ক্রমানুসারী ডিরেক্টরি কাঠামো * ২০৯
৪।।পথনির্দেশ * ২১০
৫।। ঐক্যবদ্ধ ফাইলসিস্টেম * ২১৩
৬।। ফাইলসিস্টেমের মধ্যে আলাদা আলাদা ফাইলসিস্টেম * ২১৬
৭।। ডিভাইস, ডিভাইস-ফাইল, ডিভাইস-নাম * ২১৯

।। দিন সাত।। ২২৩—২৫৮

১।। ফাইলের সম্পত্তিসম্পর্ক * ২২৩
১.১।। ফাইলের মালিকানা এবং অনুমতি * ২২৪
১.২।। ফাইলের অনুমতি বদলানো * ২২৬
১.৩।। সিস্টেম এবং ইউজার * ২২৭
১.৪।। '/etc' ডিরেক্টরির আরো দুচারটে ফাইল * ২৩১
২।। লিংকের আগে নরম করে 'cat' আর 'touch' * ২৩২

২.১।। সিম্বলিক বা সফট লিংক * ২৩৪
২.২।। সফট লিংক আর হার্ড লিংক * ২৩৬
২.৩।। আদেশের ইতিহাস * ২৩৭
২.৪।। হার্ড লিংক * ২৪১
৩।। হার্ডডিস্ক, পার্টিশন * ২৪২
৩.১।। হার্ডডিস্কের ভৌত জ্যামিতি — সেক্টর, ট্র্যাক, সিলিন্ডার, পার্টিশন * ২৪৩
৩.২।। পার্টিশন — প্রাথমিক ধারণা * ২৪৫
৩.৩।। পার্টিশনের রকমফের * ২৪৯
৪।। ফাইলসিস্টেম * ২৫২
৪.১।। বুট ব্লক এবং সুপার-ব্লক * ২৫৪
৪.২।। আইনোড-ব্লক * ২৫৫
৪.৩।। ডেটা-ব্লক * ২৫৬

।। দিন আট।। ২৫৯—২৯২

১।। ডিরেক্টরি ফাইল আর কারনেল * ২৫৯
২।। সোয়াপ পার্টিশনের স্বতন্ত্র ফাইলব্যবস্থা * ২৬২
৩।। নানা ধরনের ফাইলব্যবস্থা * ২৬৪
৩.১।। ইএক্সটিটু * ২৬৬
৩.২।। ইএক্সটিথি * ২৬৮
৩.৩।। রাইজারএফএস * ২৬৯
৩.৪।। জেএফএস * ২৭১
৩.৫।। এক্সএফএস * ২৭২
৩.৬।। আইএসও-৯৬৬০ * ২৭৩
৩.৭।। দুম্বো সাইজের ফাইল আর ফাইলব্যবস্থা * ২৭৮
৪।। তুমি যে আমার, মাউন্ট, তুমি যে আমার * ২৮০
৫।। একটু ছোট্ট কুপথ — ফাইল বদলানো * ২৮৪
৬।। গ্নু-লিনাক্স ফাইলসিস্টেম হায়েরার্কি * ২৮৬
৬.১।। রুট ডিরেক্টরি বা '/' * ২৮৭

।। দিন নয়।। * ২৯৩—৩২৬

১।। বকেয়া হায়েরার্কি * ২৯৩
১.১।। '/bin' ডিরেক্টরি * ২৯৩
১.২।। '/sbin' ডিরেক্টরি * ২৯৬
১.৩।। '/usr' ডিরেক্টরি * ২৯৮
১.৪।। '/etc' ডিরেক্টরি * ৩০৩
২।। গ্নু-লিনাক্সের কনফিগারেশন ফাইল * ৩০৬
২.১।। ডিরেক্টরি '/etc/init.d' বা '/etc/rc.d' বা '/etc/rc' * ৩০৮
২.২।। ডিরেক্টরি '/etc/sysconfig' * ৩১১
২.৩।। '/etc' ডিরেক্টরির কনফিগারেশন ফাইলেরা * ৩১৬
২.৩.১।। হোস্ট কনফিগারেশন * ৩১৬

২.৩.২ ।। '/etc/fstab', '/etc/mtab', '/proc' তথা ফাইলসিস্টেমের কনফিগারেশন * ৩২০
২.৩.৩ ।। সিস্টেম-অ্যাডমিনস্ট্রেশন বা নিয়ন্ত্রণের কনফিগারেশন ফাইল * ৩২৩
২.৩.৪ ।। সিস্টেম কমান্ডের কনফিগারেশন * ২৩৪

।। দিন দশ ।। * ৩২৭—৩৭৪

১ ।। ব্যবহারকারীর প্রোগ্রামের কনফিগারেশন ফাইল * ৩২৭
২ ।। ব্যাশ শেলের লোকাল কনফিগারেশন ফাইল * ৩৩০
৩ ।। ব্যাশের গ্লোবাল কনফিগারেশন * ৩৩১
৪ ।। অ্যালিয়াস বা ওরফ * ৩৩২
৫ ।। ব্যাশ এবং তার ভ্যারিয়েবল * ৩৩৪
৬ ।। ব্যাশ ভ্যারিয়েবলের ঠিকুজি * ৩৩৬
৭ ।। ব্যাশ ভ্যারিয়েবলের কনফিগারেশন * ৩৩৭
৮ ।। এক কুচো ব্যাশ ফাংশন * ৩৪০
৯ ।। ব্যাশ একটা স্ক্রিপ্টিং ল্যাংগুয়েজ * ৩৪২
১০ ।। ভন নয়মান আর্কিটেকচার এবং নিয়ন্ত্রণ কাঠামো * ৩৪৬
১১ ।। চিহ্নপ্রবাহ এবং তার দিকনির্দেশ * ৩৪৭
১২ ।। চিহ্নপ্রবাহের গতিপথের চালচিত্র * ৩৫০
১৩ ।। কমান্ডের মধ্যে কমান্ড * ৩৫৪
১৪ ।। অবস্থানগত নিয়ন্তা — এবং তাদের বদলানো * ৩৫৫
১৫ ।। বিশেষ ব্যাশ ভ্যারিয়েবল * ৩৫৭
১৬ ।। মেটাক্যারেকটার, গ্লব এক্সপ্রেশন * ৩৫৯
১৭ ।। কয়েকটা মেটাক্যারেকটার বা অতিচিহ্ন * ৩৬২
১৮ ।। অতিচিহ্নের থেকে নিস্তার * ৩৬৪
১৯ ।। নিয়ন্ত্রণ কাঠামো * ৩৬৫

।। সংযোজন ।। * ৩৭৫—৩৮০

১ ।। জিএলটি কী? জিএলটি কে? * ৩৭৫
২ ।। জিএলটি ইশকুল পাঠমালা — গ্নু-লিনাক্স একটি ব্যক্তিগত যাত্রা * ৩৭৮

glt-mad@ilug-cal.org

MADHYAMGRAM

গ্নু-লিনাক্স ইশকুল

জিএলটি মধ্যমগ্রাম
glt-mad@ilug-cal.org

জিএলটি-মধ্যমগ্রাম এবং এই পাঠমালা নিয়ে বেশ কিছু লেখা লিখতে হয়েছে লেখার এই আট মাসে, পাঠমালাটা লেখার নানা পর্যায়ে নানা জনের সঙ্গে নানা আলোচনার সূত্রে, নানা রকম পাঠানোর প্রয়োজনে, বাংলায় এবং ইংরিজিতে। তার থেকে দুটো এই সংযোজনে দিচ্ছি। এগুলো পাঠমালাতেই দিয়ে দেওয়ার বিষয়টায় আমি নিজে তেমন নিশ্চিত ছিলাম না। কিন্তু অন্যরা, বিশেষ করে অশেষ খুবই চেয়েছিল, এগুলো এখানে যাক। বিশেষ করে জিএলটি ব্যাপারটা নিয়ে স্পষ্টতা তৈরি করার জন্যেই। যদি অপছন্দ হয়, গালাগালটা ওকে দেবেন। আমায় নয়।

।। সংযোজন।।

১।। জিএলটি কী? জিএলটি কে?

ওদের নেট, আমাদের নেট

গ্নু-লিনাক্স তার আবির্ভাব থেকেই নেটবাহন। আমরা জিএলটি ইশকুল পাঠমালার পাঁচ নম্বর দিনে সেই আলোচনা বিশদ করে করেছি, কী ভাবে রিচার্ড স্টলম্যানের শুরু করা ফ্রি সফটওয়ার আন্দোলন থেকে তৈরি হওয়া গ্নু সফটওয়ারদের উপর দাঁড়িয়ে লিনাস টরভাল্ডসের লিনাক্স কারনেল তথা লিনাক্স অপারেটিং সিস্টেম ঘটে ওঠার জন্যেই, তার একটা অত্যাবশ্যক পূর্বশর্ত হিশেবে, ইন্টারনেটটা তার আগে থেকেই তৈরি হয়ে যাওয়ার দরকার ছিল। ইন্টারনেট না-থাকলে, গ্নু-লিনাক্সকে আমরা যে আকারে চিনি সেই আকারে গজিয়ে ওঠাটা একান্তই অসম্ভব ছিল।

কিন্তু আমরা কাজ করতে গিয়ে দেখলাম, গ্নু এবং লিনাক্সের এই সমাহার, যাকে আমি গ্নু-লিনাক্স বলে ডেকেছি এই পাঠমালায়, তার বাহন সেই নেট আর আমাদের নেট এক জিনিষ নয়। আমাদের বাস্তবতায়, আর একই কলকাতা একই বাংলার মধ্যে তো আরো অনেক কলকাতা অনেক বাংলা রয়ে যায়, তাদের অনেকগুলোর কাছেই নেট আসলে নেই। যখন নেটে টেকনিকাল প্রবেশাধিকারটুকু আছে তখন নেই নেটের সঙ্গে সাংস্কৃতিক পরিচয়, নেই ভাষাগত আত্মীয়তা। নেই নেটে চালু নানা ধরনের কায়দাকানুনের সঙ্গে আত্মীয়তা, মানে নেট আসলে নেই। যতটা নেটবাসিতা বা নেটিজেনশিপ থাকলে সেটা তৈরি হয়ে যায় একজনের মধ্যে, সে সুযোগ কজনের আছে? আমার আমাদের চেনা কলকাতায় চেনা বাংলায়?

অথচ, গ্নু-লিনাক্সের কার্যধারা — সেটা তো নেটনির্ভর। নেট থেকে আইএসও নামাও, সিডিতে পুড়িয়ে নাও, মেশিনে ইনস্টল করো, বা এফটিপি মানে ফাইল-ট্রান্সফার-প্রোটোকল সংযোগে, সরাসরি নেট থেকেই ইনস্টল করো। ইনস্টল করতে গিয়ে, বা করার পর, কোনো সমস্যা হল — লিনাক্স-ইউজার-গ্রুপে, ছোট করে যাকে লাগ বলে, সেখানে জানাও, তোমার সমস্যা সমাধানের পথ অন্য সহকর্মীরাই বলে দেবে। আর গ্নু-লিনাক্স সংক্রান্ত নানা সংবাদের জন্যে আছে নানা নিউজগ্রুপ এবং ওয়েবসাইট। নানা শাখায় ছড়িয়ে যাওয়া একটা স্নায়ুজালের মত — এর গোটাটা জুড়ে ছড়িয়ে আছে গ্নু-লিনাক্স।

কিন্তু, আমাদের ছেলেপিলেরা, একটা ফাইল ডাউনলোড করতেই, কয়েক কেবি বেশি কম হলে, যারা হিক্কা খায়, যাদের পারিবারিক বাজেট, যারা এমনকি একটা মোডেম এবং একটা নেট কানেকশনটুকুও কিনে উঠতে পারেনা, খরচ সামলে উঠতে পারেনা, অথচ, একটা কম্পিউটারে হাত মারার জায়গা একটু আছে, নিজের হোক, দাদা-কাকার হোক, কি ইন্সটিটিউট বা ওরকম কিছুর সূত্রে হোক, তারা যদি উইনডোজের সর্বব্যাপী কম্পিউটার-রাজনীতির বাইরে একটুও জায়গা খুঁজে উঠতে চায়, যেখানে অন্তত তার নিজের জানতে এবং বুঝতে চাওয়ার স্বাধীনতাটা তার নিজেরই, তখন সে কী করবে? অথচ রিসোর্সগুলো একদমই যে অপ্রাপ্তব্য তাও তো নয়, কেউ কেউ এক আধটা সিডি-ইমেজ ডাউনলোড করে, কেউ বাইরে থেকে পরিচিত কেউ আসার সূত্রে কিছু পেয়ে যায়, কেউ একটা কোনো ডিস্ট্রিবিউশন কেনে। আর গ্নু-লিনাক্সে তো যে কেউ যে কারুর কাছ থেকে এগুলো নিতে পারে, কোনো বেআইনি কাজ তাতে হয়না। তাহলে, যার কাছে যতটুকু খুঁদকুড়ো আছে সেগুলোকে এক জায়গায় মিলিয়ে ফেললেই তো বেশ একটা

রিসোর্স ভাণ্ডার তৈরি হতে পারে। রিসোর্স মানে তো শুধু সিডি না, গ্নু-লিনাক্স সংক্রান্ত নানা খবরাখবর, অন্য যারা কাজ করছে তাদের কাছ থেকে কিছু পরামর্শ। গ্নু-লিনাক্স কেন্দ্র করে যে বিপুল পরিমান বইপত্র নেটে সম্পূর্ণ অ-মূল্যে দেওয়া আছে, তারই দুটো কুচো আমি নামিয়েছিলাম, তিনটে কুচো সে, এবং চার-পাঁচটা সখা — এবার এই গোটাটা মেলালেই তো বেড়ে একটা মিনি লাইব্রেরি হয়ে গেল।

এটাই জিএলটি, এভাবেই। গ্নু-লিনাক্স ঠেক, এই কথাটাকে ইংরিজি বর্ণমালায় লিখে প্রতিটি শব্দের প্রথম আখরটা তুলে, জিএলটি। ‘ঠেক’ এই শব্দটা, জিএলটি ইশকুল পাঠমালার ‘ইশকুল’ এই শব্দটা — এই গোটাটাই সচেতন চেষ্টা, একটা বিরোধিতা, গরিব দেশের গরিব ছাগলের গরিবতর তিন নম্বর ছানাদের নেচে কুঁদে বেড়ানোর নেপথ্য দৃশ্য হিশেবে কম্পিউটারের মুলো ঝুলিয়ে, এবং একই সাথে তার সঙ্গে মহিমা এবং ভয়ের জুজু, তাদের মধ্যে ব্যাবসা চারিয়ে দেওয়ার, আর একই সঙ্গে শিখতে না-দেওয়ার রাজনীতির বিরুদ্ধে। এই রাজনীতির অংশ শুধু উইনডোজ নয়, এই রাজনীতির অংশ আমাদের গরিব দেশের দারিদ্র্যব্যবসায়ী জমিদারপ্রথার শেষ উচ্ছিষ্ট অ্যাকাডেমিবাজরাও। লোভ দেখানোর এবং শিখতে না-দেওয়ার এই দুমুখো সাপলুডোর সংস্কৃতির বিরুদ্ধে, এসো ভাই, আমাদের সাথে ঠেক মারো, এখানে ইশকুলও যা আছে তা বাঙালি জিভে প্যাঁচ খাওয়া ‘স’-বিস্তার স্কুল নয়, নিতান্তই ইশকুল, শুনলেই মনে হয়, পাশে খালপাড়ে বাঁশঝাড়ে কঞ্চি দুলছে, দেওয়ালে আবহাওয়ার নিজের হাতে রচিত জানলার ফাঁক দিয়ে প্রায়ই ঢুকে পড়ে কুকুরছানা, ইত্যাদি। কিন্তু, সেখানে চাইলে কম্পিউটার দিয়ে ভাবতে শেখার চেষ্টা করা যায়। কারণ সেটা গ্নু-লিনাক্স। উইনডোজের মত ব্যবসা নয়, কম্পিউটার অ্যাকাডেমিগুলোর মত রক্তজীবিতা নয়, ইউনিভার্সিটিগুলোর মত জোতদারমার্গ নয়। সেখানে জানতে চাইলে শেষ অব্দি জানার, শেষতম শেষ অব্দি জানার স্বাধীনতা এবং সুযোগ তৈরি করা আছে গ্নু-লিনাক্সে। কতদূর জানবে, কতদূর, কতটা বুঝবে এবং শিখবে ভাই? যতটা তোমার দম।

জিএলটি-টা আসলে এরকমই। ফর্মাল মিটিং তো আজ পর্যন্ত হয়েছে তো মাত্র দুটো। কিন্তু ছুটির দিন সকালে গুটি গুটি যে অশোকদা দেবাশিসরা আমার বাড়ির দিকে পা বাড়ায়, বা সারারাত্তির পিউর মেশিনে সুজে চটকে সকালে বাড়ি ফিরি, এগুলোই তো জিএলটি। জিএলটি বলে আসলে আলাদা কিছু নেই, আমার প্রত্যেকেই একটু একটু জিএলটি।

আমাদের কাজ

আমাদের কাজের তালিকা, যা দেখে আমরা নিজেরাই মুগ্ধ হয়ে গেছি, ও বাবা আমরা এতসব করি নাকি, দেওয়া আছে আমাদের জিএলটি সাইটে রাখা জিএলটির পরিচিতিতে।

> ১।। সফটওয়ার, ম্যানড্রেক, সুজে, স্ল্যাকওয়ার, ডেবিয়ান, রেডহ্যাট ইত্যাদি লিনাক্স ডিস্ট্রোর সিডিগুলো জোগাড় করা এবং সবার কাছে পৌঁছে দেওয়া। এদের ম্যানুয়াল, এগুলোর বই, গ্নু-লিনাক্স সংক্রান্ত অন্যান্য বইপত্র, ইলেকট্রনিক আকারে বা ছাপা অবয়বে।
>
> ২।। নিজেরা যে যেটুকু জানি সেটা দিয়ে অন্যের সমস্যাগুলো সমাধানের চেষ্টা করা। সফটওয়ারের হার্ডওয়ারের। যার সুযোগ আছে সেখান থেকে আর একটু জেনে আসার চেষ্টা করা। নিজেরাই যেটুকু জানি সেটা অন্যদের সঙ্গে মিলিয়ে নেওয়ার চেষ্টা করা, ক্লাস নেওয়া। যেরকম একটা ক্লাস থেকেই এই বইয়ের উৎপত্তি।
>
> ৩।। ওপন সোর্স সংক্রান্ত গ্নু-লিনাক্স সংক্রান্ত কম্পিউটার জগত সংক্রান্ত যে যা জানি সেটা অন্যদের কাছে পৌঁছে দেওয়া — সহজ কথায় গ্নু-লিনাক্স ঠেকবাজি। ফ্রি সফটওয়ার ফাউন্ডেশন বা লিনাক্স ইউজার গ্রুপের সঙ্গে যোগাযোগ এবং আদানপ্রদান।

এটাই আমাদের জিলটির মানে আমাদের গল্প।

আমরা কারা

আমাদের নাম এবং ঠিকানার তালিকা, ইংরিজি বর্ণমালার অনুক্রমে। এটা একদম শুরুর মিটিং-এর সময়ের সভ্যতালিকা, পরে আরো কয়েকজন যোগ হয়েছে। আরো বন্ধু হয়েছে আমাদের। তাদের নাম এখানে রইলনা।

আরাসু। ওয়ার্ড নং ১৩, সুকান্তপল্লী, বারাসাত। ফোন ২৫৬২০৫৬৬, ইমেল appal_arasu@hotmail.com।
অরিজিত মজুমদার। ৪৯, আদ্যানাথ সাহা রোড, ৭০০০৪৮। ফোন ২৫৩৪০৪২৩।
অর্ণব দেব। বারাসাত। ফোন ২৫৪২১৯৪৮।
অশেষ সেনগুপ্ত। ৩২ ডি/১, সেভেন ট্যাঙ্কস লেন, কোলকাতা ৩০। ফোন ২৫৫৬৩৯৪১।
অশোক দে। ৩ নং চণ্ডীগড়, হরিশনগর, মধ্যমগ্রাম। ফোন ২৫৩৭৫৫৬৩।
ভাস্কর দাস, উদয়রাজপুর, ম্যধ্যগ্রাম, ৭০০১২৯। ফোন ২৫৩৮৩৫৬৭, ইমেল bhaskar_laltu@yahoo.com।
দেবাশিস দাস। শ্রীনগর ২ (দুর্গামণ্ডপের কাছে), মধ্যমগ্রাম। ইমেল debuit2002@rediffmail.com।
দেবপ্রতিম দাস। ৪ নং ঝিল রোড, নিউব্যারাকপুর। সেল ৯৮৩০৪৬১২৫৫, ইমেল dpd19@indiatimes.com।
দীপঙ্কর দাশ। চন্দনগড়, মধ্যমগ্রাম, ২৪ পরগনা উত্তর, ৭০০১২৯। ফোন ২৫৩৮৫৬৩৮ ইমেল paagol@vsnl.com।
ইন্দ্রনীল দাশগুপ্ত। বি-২৭/৪, অভ্যুদয়, ইসিটিপি ফেজ ৪, কোলকাতা ১০৭। ইমেল indradg@ilug-cal.org।
ইন্দ্রনীল সান্যাল। ১৩৭/১, মসিদ হালি রোড, কালিকাপুর, বারাসাত। ফোন ২৫৬২০২৬৯।
কৌশিক দাস। উদয়রাজপুর, মধ্যমগ্রাম, ৭০০১২৯। ফোন ২৫৩৮৩৫৬৭।
কেয়া দত্ত। পেয়ারাবাগান, উদয়রাজপুর, মধ্যমগ্রাম। ফোন ২৫৩৭৩১৬০।
নিত্যানন্দ দাস। নেতাজিনগর ব্লক এ, গঙ্গানগর, মধ্যমগ্রাম, ৭০০১৩২।
প্রদীপ ব্যানার্জি। ২৪ এসপি মুখার্জি রোড, কোন্নগর, হুগলি। সেল ৯৪৩৩০৭৪৩০৮।
রাজা ঘোষ। আড়াই নম্বর গেট, বসুনগর, মধ্যমগ্রাম, ৭০০১২৯। ফোন ২৫৩৮৮৭৪৫।
শাশ্বতী গুহ। শ্রীনিকেতন, বোলপুর। সেল ৯৪৩৪০০২৪৫১।
স্মিতা ভদ্র। মদারাট রোড, বারুইপুর, ৭০০১৪৪। ফোন ২৪৩৩০৯৯৭।
তথাগত ব্যানার্জি। আরণ্যক, স্টেশন রোড, বারাকপুর, ৭০০১২০। ইমেল tathagatabanerjee@gmx.net।

আমাদের জিএলটির নিজের কোনো ভৌগোলিক ঠিকানাই নেই এখনো। ইমেল glt-mad@ilug-cal.org। সতেরোই অগাস্ট, ২০০৩-এর বিকেলে জিএলটির প্রারম্ভিক মিটিং-এ ঠিক হয়, এক বছরের মধ্যে জিএলটির কমিটি তৈরি হবে। ততদিন অব্দি কনভেনার দেবাশিস দাস, এবং রিসোর্স পার্সন অশোক দে এবং দীপঙ্কর দাশ। ইয়াহু জিওসিটিতে জিএলটির সাইট আছে, www.geocities.com/ddipankardas। সেখানে জিএলটির সব তথ্যই দেওয়া আছে, এবং আছে জিএলটি ইশকুল পাঠমালার শূন্য থেকে দশ এই এগারোটা দিনের থেকে শূন্য তিন আর চার নম্বর দিন।

২।। জিএলটি ইশকুল পাঠমালা : গ্নু-লিনাক্স — একটি ব্যক্তিগত যাত্রা

অলেখা অবই এবং গ্নু-লিনাক্স

বই একটা বিনিময়, চিন্তার। যাতে থাকে কিছু শব্দ তথা ভাষা, লাইনে লাইনে সাজানো, এবং লাইনের মাঝে মধ্যে আরো অনেক কিছু। লেখক শুধু লাইন লেখেনা, পাঠক শুধু লাইন পড়েনা। ভালো পাঠক লাইনের মধ্যে মধ্যে অলেখা অংশটা পড়তে পারে, এবং ভালো লেখক অলেখা অংশটাকে বেশি করে ভরে দিতে পারে লাইনগুলোয়। লাইনগুলো না-লিখে অলেখাটা লেখা যায়না, কিন্তু লাইন মিলিয়ে তৈরি লেখার লাইনগুলোকে অতিক্রম করে যায় অলেখাটা।

এই বইটার বেলায় কেসটা তার ঠিক উল্টো। অলেখা অংশটাই বই। আর লেখা লাইনগুলো তার কাঠামো। একটা চিত্রনাট্যের মত, শুধু আড়াআড়ি যেখানে পাট পাট করে রাখা আছে ডায়লগ এবং কাঁচা ঘটনার ছক। তৈরি হতে থাকা জ্যান্ত সিনেমার মধ্যে গজিয়ে উঠবে ওই ডায়ালগ আর ঘটনাগুলো। তাদের ছকগুলো তখন ঘটে উঠবে। এই বইটা তেমনি ঘটে ওঠার কথা একটা আগ্রহী সদ্য গ্নু-লিনাক্সীর গৃমুণ্ডের আবছায়ায়, তার নিজের মাথার ভিতরে।

এই পাঠমালাটা কোনো ম্যানুয়াল না, কী করে গ্নু-লিনাক্স শিখবেন তার ছাত্রপাঠ্য সহায়িকা না। আপনি নিজে যদি বেধড়ক আগ্রহী হন গ্নু-লিনাক্স শিখতে, না, তার চেয়েও বেশি, আপনি যদি আগ্রহী হন গ্নু-লিনাক্স দিয়ে ভাবতে, তার জন্যে লড়ে যাওয়ার বাসনা থাকে আপনার, বেমক্কা পরিশ্রম করার বাসনা, তাহলে, সেই ম্যারাথন দৌড়ের সময় মোড়ে মোড়ে আপনার হাতে জলের কৌটো এগিয়ে দেওয়ার একটা চেষ্টা। বা ধরুন গ্নু-লিনাক্স শেখার বক্সিং-এর রিঙে রাউন্ডের ঘন্টা পড়া মাত্র, আপনি চেয়ারে গিয়ে বসা মাত্র, আপনাকে ভেজা স্পঞ্জ এগিয়ে দেওয়া, বা একটু পায়ের গুলি মালিশ করে দেওয়া। বইটা আসলে লিখছে গ্নু-লিনাক্স, আগ্রহী গ্নু-লিনাক্সীর মুন্ডুর মধ্যে, এই পাঠমালা নামের এই অবইটা তার চিত্রনাট্য।

একদম আলাদা একটা কাঠামো

একটা শহর চেনার প্রক্রিয়ার প্রতিতুলনাটা নিলে অন্য বইয়ের কাঠামোর সঙ্গে এর কাঠামোর ফারাকটা বোঝা সহজ হবে। ধরুন একজন সিরিয়াস গম্ভীর বর্ষীয়ান অ্যাকাডেমিক তার শহরে আসা একজন আগন্তুককে শহর চেনানোর দায়িত্ব পেলেন। তিনি কী দিয়ে শুরু করবেন? পৃথিবী নামক গ্রহের ভূগোলের নিরিখে শহরটার অবস্থিতি তথা ভৌগোলিক খুঁটিনাটি চেনাবেন, শহরটার ইতিহাস জানাবেন, তারপর শহরটার উপর করা সমাজবৈজ্ঞানিক নানা পাঠের সঙ্গে পরিচিত করাবেন বহিরাগতকে, শহরটার রাজনৈতিক সাংস্কৃতিক জীবন — ইত্যাদি সবকিছুতে, ক্রমে ক্রমে। একটা অভ্যস্ত সঙ্গত অ্যাকাডেমিক টেক্সটবই এভাবেই গড়ে ওঠে। একটা যুক্তিবৈজ্ঞানিক বা লজিকাল ছক, তার মধ্যে একটু করে রক্তমাংস ভরে ভরে শহরটার ধারণা গড়ে তোলা হবে আগন্তুকের মাথায়, তারপর বাস্তব শহরের নানা ঘটমানতার সঙ্গে তার সেই জানাটাকে মিলিয়ে দেখার জায়গা বানানো হবে।

কিন্তু ধরুন, এই দায়িত্বটা পেল বাজে-বকা বেশি-বকা কোনো ভুলভাল ঠেকবাজ। সে-ও তো চেনাবে শহরটাকে। কী ভাবে?

সে প্রথমেই আগন্তুককে নিয়ে ঘুরতে মানে ঠেক মারতে বেরোবে, নিজেও রোজ যা করে। বেরিয়ে নিজের খুশি মত ঘুরতে থাকবে শহরের রাস্তায়, যে রাস্তায় আগন্তুকের সঙ্গে মিলে ঘুরে বেড়ানোটা তার ইন্টারেস্টিং লাগছে। যে গলি, উপগলি, কানাগলি বেয়ে সে আগে কখনো গেছে বা যায়নি। কোথাও যেতে যেতে কোনো দোকানের সামনে দাঁড়িয়ে পড়বে, কোথাও একটা বড় গর্তের সামনে, এখানেই একটা কামানের গোলা পড়েছিল, কোথাও একটা রেলিঙের সামনে — শহরের যুবক-যুবতীরা এখান থেকেই সুইসাইড খায়, প্রেমের মাঠের ঘাস খাওয়ার সুযোগ না-থাকলে। যেখানে তার মনে হল, আগন্তুকের চোখটা যেন একটু চিকচিক করছে, একটু যেন বেশি আগ্রহ, সেখানে একটু বেশি বাজে বকবে। এরকম করতে করতে যখন মনে হবে একদিনের মত করে যথেষ্ট ঘোরাঘুরি হয়েছে, তখন এক দিনের মত শহরচেনা সাঙ্গ করে তারা আস্তানায় ফিরে যাবে। আবার পরের দিন বেরোতে হবেনা? এখান থেকেই এসেছে 'চ্যাপ্টার'-এর বদলে 'দিন'-এর ধারণাটা। বইয়ের চ্যাপ্টার ঘটে বইয়ের বিষয়ের আভ্যন্তরীন যুক্তির বিকাশের সঙ্গে

মিলিয়ে। এখানে কোনো যুক্তিবৈজ্ঞানিক যৌক্তিকতা নেই, একদমই মেঠো যুক্তি — এই আমার ঠ্যাং টনটন করছে রে, আবার কাল, হ্যাঁ?

এই রকম ঘোরাঘুরি আর বকাবকি করতে করতে যখন তার নিজের ভাঁড়ার খতম হয়ে যাবে, এবং সেটা প্রায়ই যাবে, বা আর বকবক করতে ইচ্ছে করবে না, তখন সে দেখিয়ে দেবে, ওই যে লোকটাকে দেখছেন, ওর বড়মামা এই শহরের পত্তন করেছে, ওর সেজবৌদি শহরের কাউন্সিলর, ও একটা ডাইরেক্টরি লিখেছে, সবার পড়ার জন্যে রেখে দিয়েছে লাইব্রেরিতে, সেটার থেকে পড়ে নিন ভাই, আমি একটু বুদ্ধির গোড়ায় ধোঁয়া দিয়ে আসি। জিএলটি ইশকুল পাঠমালায় একদম এই কায়দায় মুহুর্মুহু পাঠককে ঠেলে দেওয়া হয়েছে গ্নু-লিনাক্স সিস্টেমের মধ্যেই দেওয়া থাকা পাঁজা পাঁজা ম্যানুয়ালের দিকে। বা কোথায় গিয়ে কী করলে নিজেই এর মানেটা খুঁজে নিতে পারবেন সেই দিকে। প্রায়ই প্রায় কিছুই বলা হয়নি। যান ভাই পড়ে নিন, অত বকতে পারছি না, আর অত ভালো করে বলতেও পারব না, আমি তো ম্যানুয়াল লিখিয়ে নেই, আমি গ্নু-লিনাক্স শহরের একজন আয়েসি ঠেকবাজ মাত্র। নিজেই দেখে নিন না। আপনাকে তো নিজেই চিনতে হবে, নিজের ঠেক বানিয়ে নিতে হবে, আমি তো নিমিত্ত মাত্রও নই।

এবং, একজন ঠেকবাজের তো কোনো দায় নেই। ইতিহাসগত-ভাবে ভূগোলগত-ভাবে মানববিজ্ঞানগত-ভাবে সামগ্রিক সঙ্গত সঠিক এবং সুপ্রযুক্ত হওয়ার। তার নিজের যেসব জায়গায় যেতে ভালো লাগে, যাদের সঙ্গে ঠেক মারতে ভালো লাগে, সেখানে সে আগন্তুকটিকে ঘুরিয়ে বেড়াচ্ছে মাত্র। এইখানটায় দাঁড়িয়েই পাঠকের পাঠের চরিত্রটা অন্য বইয়ের থেকে একদম পালটে যাচ্ছে। পাঠক আর গ্নু-লিনাক্স বা ওই সংক্রান্ত কোনো লেখা পাঠ করছে না। পাঠ করতে শুরু করছে এই ঠেকবাজের শহর চেনার রকমটাকে, এই শহরের গলি-উপগলির সঙ্গে লোকগুলোর সঙ্গে ঠেকগুলোর সঙ্গে এই ঠেকবাজের জ্যান্ত সম্পর্কটাকে। মানে পাঠ করছে লেখকের সঙ্গে গ্নু-লিনাক্সের সম্পর্কটাকে। পাঠক নিজেই এবার একটা জ্যান্ত রকমে পড়তে শুরু করছে, কিন্তু সেই পাঠটা এই লেখাটার নয়, গ্নু-লিনাক্স নামে জ্যান্ত শহরটার একজন জ্যান্ত বাসিন্দাকে, মানে জ্যান্ত গ্নু-লিনাক্সকে।

তাই, কোনো বই লেখা হচ্ছেনা, ঠেকবাজ আর আগন্তুকের কথাবার্তা, ফিসফাস, চুপ-করে-থাকা, চ্যাংড়ামি — এই সব কিছুকে মিলিয়ে গজিয়ে উঠছে শহর চেনার প্রক্রিয়াটার একটা চিত্রনাট্য। শহরটা ক্রমে গজিয়ে উঠতে শুরু করছে আগন্তুকের মাথার ভিতরে। তাই, কোনো নিখাদ পাঠক এই অবইয়ের কোনো পাঠকই নয়। পদব্রজী ঠেকব্রজী আগন্তুক হতে হবে তাকে, হেঁটে দেখতে শিখুন, ঝরছে কী খুন দিনের রাতের মাথায়। তার মাথার মধ্যে মিলে যাওয়ার এই প্রক্রিয়াটা একটা নিয়ন্ত্রিত এলোমেলোপনা, কন্ট্রোলড ক্যাওস। আমার নিজের শহর চেনার স্মৃতি থেকে যে গলিগুলো খুব জটিল লেগেছিল, যে এলাকাগুলো চেনা এবং বোঝা খুব প্রয়োজন মনে হয়েছিল, সেগুলো দিয়েই বারবার নিয়ে যাচ্ছি আমি। এক একবার এক এক দিক থেকে। এবং অ্যাকাডেমিক একটা টেক্সটবইয়ের সঙ্গে একটা বড় ফারাক এখানেও। সেখানে একটা নিয়ম মেনে, আগের পরের বিষয়ের সঙ্গে যুক্তিবৈজ্ঞানিক সম্পর্ক রেখে প্রতিটা বিষয় আসে। এখানে এসেছে একজন ব্যক্তিগত ঠেকবাজের ব্যক্তিগত যাত্রার অভিজ্ঞতাপ্রসূত গুরুত্বের নিয়ম মেনে, ভালো-লাগার নিয়ম মেনে, সুবিধের নিয়ম মেনে।

কাদের জন্যে লেখা

এ জাতের কোনো বই কেউ কোথাও দেখেছে শুনলেই শ্রোতা প্রথম প্রশ্নটা করে, আরো গ্নু-লিনাক্স নিয়ে চারদিকে হঠাৎ গজানো মিডিয়া উৎসাহের হিড়িকে, ‘হ্যাঁরে, ওটা পড়লেই লিনাক্স শেখা যাবে?’ ঔপনিবেশিক নেটিভ যেমন করে ধরে নেয়, ইংরিজি শিখে ফেললেই, পাউডার মেখে ফরসা হলেই সাহেব হতে পারবে, মেমের পাশে বসে অর্কেস্ট্রা দেখতে পাবে প্যান্টুলুন পরে — এই ফ্যান্টাসিতে ভোগে। তেমনি আর এক ব্রান্ডের সাহেবতার হিড়িক।

তাদের মৃদু হেসে বলার, একদমই না, এই অবইটা পড়ে আপনি যে কিছুতেই গ্নু-লিনাক্স শিখবেন না, এটা একদম শূন্য নম্বর দিনের শুরুর মুহূর্তেই বলে নেওয়া আছে, পুরো দস্তুর অনসাইট ওয়ারান্টি সহ। এই জাতের পাঠকের মুখ ম্লান করে দেওয়া ছাড়া প্রিয় কাজ নেই এই অবইয়ের। যে পাঠক নিজের ক্রিয়াকে পড়তে শেখেনি, তাকে রসগোল্লা খিলাতে খিলাতে কোনো কিছু শিখিয়ে ফেলার ধৃষ্টতাই নেই এই অবইয়ের। যন্ত্র কিনুন, ম্যানুয়াল ফ্রি — সেই গোত্রের কোনোকিছু যাতে কিছুতেই না-হয়ে পড়ে এই পাঠমালাটা, তার দিকে পুরোদস্তুর মনোযোগ দেওয়া হয়েছে। লেখক হিশেবে পাঠমালাটার বিক্রির নিরিখে সবচেয়ে স্বাস্থ্যকর যা বলতে পারি, তাও একটু কুঁতে, কোনো সুশ্রাব্য

বাক্য শুনলেই কোঁত পাড়ার বয়সে পৌঁছে গেছি অনেকদিন হল — কেউ যদি নিজেই গ্নু-লিনাক্সের প্রক্রিয়ার অংশীদার হওয়ার ব্যক্তিগত যুদ্ধে নামতে চায়, তার সবচেয়ে নিকট সহযোদ্ধা হওয়ার চেষ্টা করেছে অবইটা।

জিএলটিতে আমার চারপাশে, আমার চেয়ে বয়সে অনেক ছোট যেসব বন্ধুদের সঙ্গে গ্নু-লিনাক্স নিয়ে কথাবার্তা বলেছি, নানা প্রশ্ন উঠেছে, বা পড়াতে গেছি যখন জিএলটির ক্লাসে, সেখানে যে জিনিষগুলোকে দেখেছি সবচেয়ে বেশি ছেলেপিলেদের উত্তেজিত করে, যে প্রসঙ্গগুলো দিয়ে গ্নু-লিনাক্সে ভাবতে পারাটা সবচেয়ে বেশি চারিয়ে দেওয়া যায়, তাদেরকেই নিয়ে আসার চেষ্টা করেছি লেখাটায়। উইনডোজ করতে করতে অন্তরালে রয়ে যায় মেশিনের যে দানবিক শক্তি, সেটাকে নিজের মত করে যে একবারো অনুভব করেনি, করতে চায়নি, দৈত্যদের সঙ্গে টেবিল টেনিস না-খেলে উইনডোজে মাউসের পেটে কাতুকুতু দিয়ে ‘বঃ, এই তো বেশ কম্পুটার জানি’ ভেবে ফেলাতেই, নিজের পিঠ চুলকে নেওয়াতেই যার রুচি, তার জন্যে এই লেখাটা কিছুতেই লেখা হয়নি। কারন বুঝতে বা জানতে চায়নি সে নিজেই, শুধু নিজের ঝুলে-যাওয়া কনফিডেন্স খাড়া করতে চেয়েছে, আয়নার দিকে তাকিয়ে যাতে বলতে পারে, ওয়ালা : এক্সে হোমো — এই দেখো লোকটার কোট, এই দেখো গলার জাঙ্গিয়া, কী স্মার্ট দেখাচ্ছে, লোকটা আবার লিনাক্স-ও জানে। তার মনোরোগের দাওয়াই নয় এই পাঠমালাটা।

জিএলটি মধ্যমগ্রামের তরফে ত্রিদিব সেনগুপ্ত (দীপঙ্কর দাশ)

glt-mad@ilug-cal.org

MADHYAMGRAM

গ্নু-লিনাক্স ইশকুল

জিএলটি মধ্যমগ্রাম
glt-mad@ilug-cal.org

আমরা আমাদের গ্নু-লিনাক্স ইশকুলের পড়াশুনো শুরু করতে যাচ্ছি — লিনাক্স শিখে ফেলব এখানে, তা নয়। লিনাক্স কী ভাবে শিখব সেটা শেখার চেষ্টা করব। আজ আমরা আসলে সেটাও করব না, এটা শূন্য নম্বর দিন, পরের এক নম্বর দিন থেকে আমরা যেগুলো শুরু করব তার একটা ভিত খুঁড়ে নেওয়ার চেষ্টা করব আজকে। যে কোনো একটা বড় সক্রিয় তীব্র আর জ্যান্ত কাজের বা বিদ্যার এলাকায় যা হয় — শেখাটাই হয়ে দাঁড়ায় নিজের একটা গবেষণা। প্রত্যেকটা ছাত্র আসলে তার নিজস্ব স্বতন্ত্র একখানা বিদ্যা শেখে। কিন্তু এই শেখাটায় পৌঁছনোর জন্যে কিছু মকশো করা লাগে, কিছু বাঁদরের মুখ ভ্যাংচানোর মত নকল, কিছু হাতড়ানো, কিছু লাগে-তাক-না-লাগে-তুক, কিছু ব্রুট ফোর্স মানে গাজোয়ারি, কিছুটা নার্ভাস হয়ে যাওয়া — বেশ কয়েক কোটি বার মনে হওয়া — ধুর এতো শালা কিছুই পারছি না। তারপর হঠাৎ একদিন আবিষ্কার করা, আরে, বেশ তো পারছি। কাজটা তো কেমন হয়ে গেল। জানিনা, এর থেকে অন্যরকম, এর থেকে ভালো কোনো শেখা থাকতে পারে। কিন্তু আমি সেখানে কাউকে কোনো পরামর্শ দিতে পারব না। কারণ, আমি সেটা জানিনা, আমি নিজে যেরকম ভাবে করেছি, উত্তেজিত হয়েছি, বারের পর বার ধেড়িয়েছি, তারপর আকস্মিকভাবে মিলে গেছে, আমি ঠিক সেটাই রাখছি এখানে, যদি অন্য আর কারুর এতে কোনো সাহায্য হয়। শুধু যে আমি কোনদিন প্রথাগতভাবে সত্যিকারের ঠিক মতো উপায়ে কম্পিউটার শিখিনি তাই নয়, কম্পিউটার আমার নিজের এলাকাও না, আমার কাছে প্রায় এক দশক আগে কম্পিউটার এসেছিল একটা ঝিনচাক টাইপরাইটার হয়ে — কোথাও জানতাম যে এতে লেখার, লিখে চলার এবং সেই লেখা অন্যকে পড়ানোর গোটা প্রক্রিয়াটাই সহজতর হয়ে যাবে। তখন শুরু করেছিলাম উইনডোজে। তারপর অল্প কিছু দিন হল আমার গ্নু-লিনাক্সে আসা। যে অভিজ্ঞতাটা আমার কাছে এত নাটকীয় এবং উত্তেজক যে, মাঝে মাঝেই মনে হয়, এত পরে কেন, এত বেশি বয়সে কেন? আগে হলে কী ভালো হত। তাই, আমার তৈরি করা নোট পড়াটা সত্যিকারের ভালো কম্পিউটার শেখার পক্ষে কতটা ভালো সে বিষয়ে আমার নিজেরই সন্দেহ আছে। সামনাসামনি যারা জিএলটিতে আসছে তারা তবু বুঝে নিতে পারে — কতটা ধরবে আর কতটা ছাঁটবে। ছাঁটার প্রয়োজন থাকবে প্রচুর — সেই স্কুলজীবনেই শিলাদিত্য বলেছিল — একটা লোক এত উৎকেন্দ্রিক হয়? (ও সাংস্কৃতিক ছেলে বলে একসেনট্রিক বলত-না) আমার সেই কেন্দ্র থেকে উৎক্ষিপ্ততা তো আমার লেখা টেক্সট-এ থাকবেই, গত বিশ বছরে আমার প্রত্যেকটা লেখাতেই থেকেছে। প্লিজ, পড়ে ক্ষিপ্ত হবেন না। বরং, লিনাক্সের যেটা চালু প্রথা, কমিউনিটিকে ফিরিয়ে দাও তুমি যা পেয়েছ, সত্যিকারের ভালো কেজো এবং শিক্ষিত একটা লেখা লিখে ফেলুন। আপনারা তো অনেকেই আছেন যাদের কাজের এলাকাই কম্পিউটার — কতো ভালো করে জানেন, একদম ভিতর থেকে, আপনাদের কাছে কাজটা আরো অনেক সহজ। আমার কাছেও গ্নু-লিনাক্স-এর সবচেয়ে গুরুত্বপূর্ণ জায়গা এটা — আমাদের মত গরিব দেশে গ্নু-লিনাক্সটা প্রথম বিশ্বের চেয়ে আরো হাজারগুণ জরুরি। ট্রেন ইস্টিশনে যখন লোককে শুয়ে থাকতে দেখেন খালি পেটে, হাসপাতালে দেখবেন লোকে বেড পাচ্ছেনা, চাঁদের মত মুখ নিয়ে ছবছরের বাচ্চা রাস্তার পাশে চারবছরের বাচ্চাকে কোলে নিয়ে ভিক্ষে করছে — তখন নিজেকে মনে করাবেন কথাটা।

।। দিন শূন্য ।।

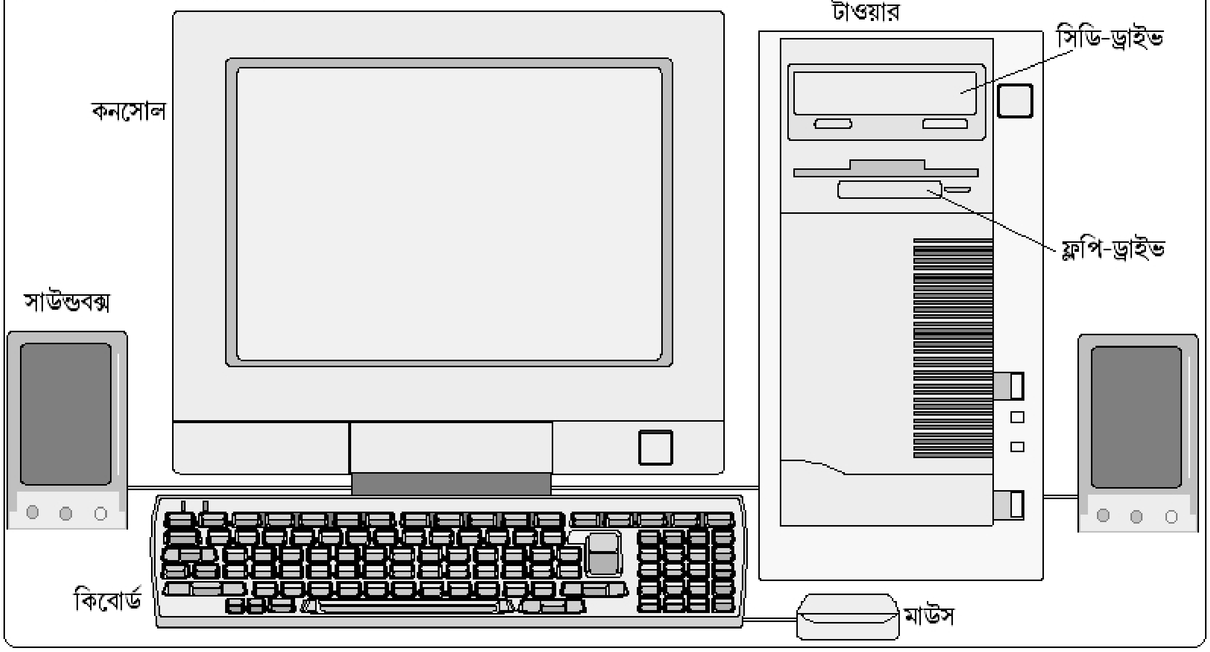
০.০ ।। সামনে একটা আস্ত পিসি

সবার হয় কিনা জানিনা, প্রথমবার সামনে একটা আস্ত পিসি দেখে আমার মাথা খারাপ হয়ে গেছিল। এটা একটা কম্পিউটার — এটা নিয়ে এখন আমি যা-খুশি করতে পারি? এর পরের স্টেপে আসে — কোন যা-খুশি-টা করা যায় কোনটা যায়না — সেটা বোঝার চেষ্টা করা। এইখানে প্রথম যেটা দরকার পড়ে — একটা কম্পিউটারের গঠনটাকে বোঝা। আমরা এই প্যারাগ্রাফে পিসি মানে পার্সোনাল-কম্পিউটার আর কম্পিউটার সমার্থক হিশেবে ব্যবহার করছি, কিন্তু পরে আমরা দেখব এখানে আরো অনেক অষ্টাশি মজা আছে।

কম্পিউটারটা পার্সোনাল কি ইম্পার্সোনাল কি সোশাল কি অ্যান্টিসোশাল সেটা বড় কথা না, পার্সোনাল কম্পিউটার একটা বিশেষ ধরনের কম্পিউটারের নাম, এবং আমাদের বেশিরভাগেরই কম্পিউটারে হাত-পাকানো শুরু হয় এই মেশিনেই। আসুন আমরা একটা আমাদের প্রতিদিনের অভ্যস্ত পিসির গঠনটার সঙ্গে অভ্যস্ত হই। ঠিক যেরকম আমরা টেবিলের ওপরে দেখি। এর দৃশ্যমান অংশগুলোকে আর একবার খেয়াল করে নিই। দুপাশে দুটো সাউন্ডবক্স। সামনে কিবোর্ড আর মাউস। পাশে টাওয়ারটা। যার সামনে থেকে দেখা যাচ্ছে সিডি-ড্রাইভ আর ফ্লপি-ড্রাইভটা, আর দু-একটা আলো বা কোনো ছোটখাটো ডিসপ্লে। আর প্রায় সব মেশিনেই সামনে দু-একটা সুইচ, যার একটার পাশে অনেকক্ষেত্রেই লেখা ‘রিসেট’। আর সবার উপরে মাথা তুলে আছে কনসোল। আমার দেড়বয়স্ক ভাগ্নে বিরক্ত হয়ে বলেছিল — তোমার টিভিতে সিনেমা হয়না? হয় যে, সেটা আর আমি বলিনি, এখনো আমি অত পাগল নই।

মজার কথা কি বলুন তো — আসলে কম্পিউটারটা কিন্তু অদৃশ্যই হয়ে রয়েছে — প্রায় পুরোটাই। ধরুন, আপনার একটা একটা করে আঙুল চুল ঠোঁট জামা গলার শব্দ স্বপ্ন সব ছেঁটে ফেলা হচ্ছে — কতক্ষণ অব্দি আপনি আপনাকে ‘আপনি’ হিশেবে টিকিয়ে রাখতে পারেন — তার একটা এক্সপেরিমেন্ট। কোন কোন পার্টটুকুতে কেন্দ্রীয়তম ‘আপনি’ বসবাস করেন সেটা আমরা খুঁজছি। আপনার ক্ষেত্রে এই কাটাকাটিটা যতটা জটিল আপনার পিসির বেলায় তা-নয়। তাকে পার্টে পার্টে পিসপিস করে ফেলা যায়, পরে আমরা একটু আধটু সেটা করব-ও। সেই পার্টগুলোর ভিতর সবচেয়ে জরুরি যেটা পাব সেটা প্রসেসর মানে মাইক্রোপ্রসেসর চিপ। এটা থাকে টাওয়ারের ভিতর, মাদারবোর্ডের গায়ে সাঁটা। এই মাদারবোর্ড সত্যিই সবগুলো আভ্যন্তরীন অংশেরই মাদার, সবাইকেই ধরে রাখে। মাদারবোর্ডের গায়েই সাঁটা থাকে কম্পিউটারের দ্বিতীয় জরুরিতম অংশ — মেমরি বা স্মৃতি। এর পর থাকে হার্ড ডিস্ক, যাকে বলাই যায় যে কম্পিউটারের তৃতীয় জরুরিতম অংশ। এছাড়া টাওয়ারের মধ্যে থাকে বিভিন্ন অ্যাডাপ্টার সহ অ্যাডাপ্টার কার্ড। যা যা দিয়ে আপনার পিসি জন্মেছিল, যে সমস্ত হাত-পা-চুল-লেজ ইত্যাদি, তার বাইরে আরো নানা হাত-পা-চুল-লেজ মায় মুন্ডুও অ্যাডাপ্ট করে, লাগিয়ে নেয় নিজের সঙ্গে এই অ্যাডাপ্টার কার্ড দিয়েই। ইস, আমাদেরও যদি এরকম হত। বিশেষ করে একটা সাপ্লিমেন্টারি মুন্ডুর প্রয়োজন প্রায়ই এত বোধ করি। টাওয়ারের মধ্যে এছাড়াও থাকে বৈদ্যুতিক যোগাযোগের একটা গম্ভীর কালো চৌকো বাক্স — ধাতুর পাতের। গ্রীক পুরানের মেডুসার মাথার মত যার থেকে বেরিয়েছে কোটি কোটি তার। আর থাকে দুটো বা তারো বেশি ফ্যান — শোঁ শোঁ আওয়াজ করে, মেশিন চলে যখন।

একটা কম্পিউটারের গঠন



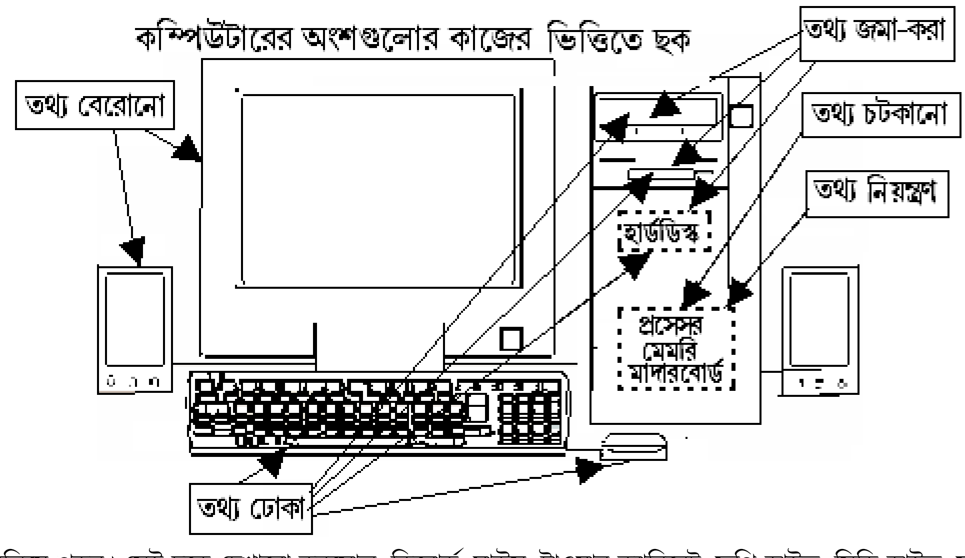
বাইরে থেকে যেগুলো দেখা যাচ্ছে, তার মধ্যে খুব জরুরি হল কনসোল আর কিবোর্ড। কিবোর্ড থেকে কম্পিউটারে নানা আদেশ পাঠাতে হয় — নইলে ব্যাটা চলবে কী করে। আর কনসোলে ফুটে ওঠে সেই আদেশগুলো এবং সেই আদেশ পাওয়ার পর কম্পিউটার যা যা করল তার ফলাফল। এই কিবোর্ডেরই একটা বিকল্প মাউস বা ইঁদুর। আমরা কিবোর্ডে আদেশ দেওয়ার বদলে যখন মাউস দিয়ে আদেশ দিই, তখন, পিছনে, আমাদের অগোচরে মাউস দিয়ে দেওয়া আদেশগুলোকে বদলে ফেলে কম্পিউটার, কিবোর্ড থেকে দেওয়া আদেশের আকারে নিয়ে আসে। সেই ব্যবস্থা করা থাকে কম্পিউটারে। পিছনে পিছনে মাউসের আদেশগুলোকে সে কিবোর্ডে টাইপ করা আদেশের মতো চেহারায় অনুবাদ করে নেয়, রূপান্তরিত করে নেয়। পরে আমরা একটু একটু করে বুঝব এসব।

এর বাইরে যেটা এই ছবিতে আছে তা হল সাউন্ডবক্স দুটো। এখন এটা পিসির বেলায় প্রায় একটা মাস্ট হয়ে গেছে। গান শোনার, মাল্টিমিডিয়া চালানোর জন্যে। মাল্টিমিডিয়া বলতে মাল্টিপল মিডিয়াম যেখানে একসঙ্গে চলছে, নানা ধরনের ফাইল, তৈরি করছে ছবি আর শব্দ। ছবিটা আসছে কনসোলে। আর শব্দটা এই সাউন্ডবক্সে। যেমন এটা লিখছি আর সাউন্ডবক্সে বাজছে দিলীপ রায়ের ‘পুষ্প উছল গাহে কাননে কোকিলা, শ্রাবণবাদল ছায় নভে’, আর এইসব বেজার লেখা লিখতে লিখতে ভাবছি কখন এসব থামিয়ে হিন্দুস্তানির ‘টেলিফোন-ধুনমে হাঁসনেওয়ালি’ দেখতে পারব — তখন গানের সঙ্গে সঙ্গে চলবে কমল হাসান আর মনীষা কৈরালার চলৎছবি, শঙ্করের দুর্ধর্ষ ক্যামেরায়।

এই যে অংশগুলোর কথা বললাম এগুলোর প্রত্যেকটারই নিজের নিজের কাজের ভিত্তিতে কম্পিউটারের মধ্যে একটা নিজস্ব ভূমিকা আছে। এবার কম্পিউটারের সেই কাজের ভিত্তিতে গঠনটা বোঝার চেষ্টা করা যাক। গোটা কাঠামোটা একটু একটু করে আমাদের কাছে ফুটে উঠবে, এই একই ছকের আলোচনায় আর একবার ফিরে আসব আমরা এই পাঠমালার শেষ মানে দশ নম্বর দিনে, মানে এগারো নম্বর চ্যাপ্টারে, ভন নয়মান আর্কিটেকচারের আলোচনায়।

০.১।। কাজের ভিত্তিতে একটা কম্পিউটারের গঠন

এই চিত্রে কম্পিউটারের অংশগুলোকে তাদের কাজের ভিত্তিতে দেখানো হয়েছে — আগের পার্সোনাল কম্পিউটারের ছকের



সঙ্গে মিলিয়ে পড়ুন। সেই ছকে দেখানো কনসোল, কিবোর্ড, মাউস, টাওয়ার ক্যাবিনেট, ফ্লপি ড্রাইভ, সিডি ড্রাইভ, সাউন্ড-বক্স ছাড়াও এখানে সচ্ছিদ্র দাগের ভিতর দেখানো হয়েছে প্রসেসর, মেমরি, মাদারবোর্ড এই তিনটেকে একসঙ্গে — যারা তথ্য চটকানো বা প্রসেসিং এবং নিয়ন্ত্রণ বা কন্ট্রোল — এই দুটো কাজে অংশগ্রহণ করে। অন্য সচ্ছিদ্র দাগের মধ্যে দেখানো হয়েছে হার্ডডিস্ককে, যে অংশ নেয় তথ্য জমা-করা এবং তথ্য ঢোকানো — এই দুটো কাজে। এই দুটো সেটকে সচ্ছিদ্র দাগের ভিতর দেখানোর কারণ এই যে এদের বাইরে থেকে দেখা যায়না। আগের ছবির সঙ্গে মেলান। তথ্য ঢোকানোর কাজ করে কিবোর্ড, মাউস, ফ্লপি ড্রাইভ, সিডি ড্রাইভ আর হার্ডডিস্ক ড্রাইভ। তথ্য বার করার কাজ করে কনসোল আর সাউন্ডবক্স। এখানে আমরা হাতে গোনা এই কয়েকটা মাত্র উপাদান দেখিয়েছি। বাস্তবে সম্ভাব্য উপাদানের সংখ্যা তো হাজারে হাজারে।

একটা কম্পিউটারকে কম্পিউটার-সমাজে কম্পিউটার হিশেবে মুখ দেখাতে গেলে কিছু কেজো অংশ থাকতেই হবে। কাজের যুক্তির ভিত্তিতে এই গঠনটা এইরকম —

১।। ডেটা ইনপুট বা তথ্য ঢোকা। মানে তথ্য এবং আদেশ মেশিনে ঢোকা। আপনি আপনার চালের এবং ডালের এবং সিগারেটের খরচ কম্পিউটারে ঢোকালেন — এটা তথ্য। তাদের যোগ করতে বললেন, স্ক্রিনে দেখাতে বললেন — মাস থেকে মাসে তাদের চলমান হিশেবটা — এটা আদেশ। তথ্য বা আদেশ ঢোকানোর কাজটা করলেন কিবোর্ড মাউস ইত্যাদি দিয়ে। সেগুলো হল ইনপুট ডিভাইস, তাদের কাজ কম্পিউটারে তথ্য ঢোকানো। পরে আমরা দেখব, শুধু এই দুটো ডিভাইসই নয়, অন্য ডিভাইসও এই কাজে ব্যবহার করা যায়। কাজ এবং ডিভাইস বা উপাদানগুলোর মধ্যে প্রায় কোনো ছুতমার্গই নেই, পরে একটা টেবিল করে দেখব আমরা, প্রায় সব ডিভাইসই একের বেশি কাজে ব্যবহার করা যায়।

২।। ডেটা আউটপুট বা তথ্য বেরোনো। কম্পিউটার কনসোলে কিম্বা প্রিন্টারে প্রিন্টরত পাতায় হিশেবটা বেরিয়ে এলো। যা বেরিয়ে এল সেটা তথ্য। যেখানে বেরিয়ে এল মানে সেই কনসোল বা প্রিন্টার হল আউটপুট ডিভাইস। যদি আউটপুট একটা ফ্লপিতে লিখে নেন, বা একটা সিডিতে পুড়িয়ে নেন, তখন সেটাই হয়ে দাঁড়াল মেশিন থেকে তথ্য বার করার আউটপুট ডিভাইস।

৩।। ডেটা প্রসেসিং বা তথ্য চটকানো। আমার ঢোকানো তথ্যগুলোকে আমারি ঢোকানো আদেশ মোতাবেক চটকানো বা প্রসেসিং — এটাই কম্পিউটারের মোদ্দা কাজ। খেয়াল করুন। পরে, তিন আর চার নম্বর দিনের আলোচনায় দেখবেন, কম্পিউটার ব্যাপারটাই গজিয়ে উঠেছে এই সংখ্যা চটকানোর কাজের বিবর্তনকে ঘিরে। এটাই কম্পিউট করা মানে হিশেব করা, যা কম্পিউটারকে কম্পিউটার নামটা দিয়েছে। আর সব কাজই হল এই কাজটার জুতো টুপি টাই ইত্যাদি, মানে এটা করতে গিয়ে আর যা যা লাগে। যদিও এখন, আজকের পৃথিবীতে, কম্পিউটারে হিশেব করার কাজ প্রায় অনুপস্থিত বললেই হয়। মোট জ্যান্ত মেশিনের ক-পিস আর ব্যবহার হয় সংখ্যা চিবোনোর কাজে? বরং তথ্য সম্প্রচার বা ডেটা কমিউনিকেশন আর সেই কমিউনিকেশনের জন্যে দরকারি তথ্য তৈরি করা, ডেটা জেনারেশন, এতেই ব্যবহার হয় বেশিরভাগ মেশিন। আপনি একটা প্রবন্ধ লিখলেন, পড়তে পাঠালেন লোককে ইমেল করে — গোটাটাই তথ্য সম্প্রচার, আবার ইন্টারনেট থেকে ডাউনলোড করে আপনি একটা মুভি ক্লিপ দেখলেন, তার মানেও তথ্য সম্প্রচার। কিন্তু হিশেব মানে সংখ্যা নিয়ে কাজ করা থাকুক আর না-থাকুক, কম্পিউটারের মূল কাজই হল তথ্য চটকানো বা ডেটা প্রসেসিং। এর থেকে মুক্তি নেই বেচারার। প্রোগ্রাম মারফত আমার দেওয়া আদেশে আমার ঢোকানো তথ্য চটকানো। মানে আমারি দেওয়া শিলে আমারই দেওয়া নোড়া, কম্পিউটার এবার পিন্ডি চটকালো — নইলে ওই পিণ্ড পিণ্ড তথ্য বেরিয়ে আসবে কী করে কনসোলে বা প্রিন্টারে।

৪।। ইনফর্মেশন হোল্ডিং বা তথ্য জমানো। তথ্য যদি চটকাতে হয়, তাহলে তথ্য কোথাও, যতটুকু সময়ের জন্যেই হোক, তাকে তুলে রাখতে হবে। অনুপস্থিত তথ্য আপনি চটকাবেন কী করে? এই তুলে রাখাটা অল্প এককুচি সময়ের জন্যে হতে পারে, চোখের পলক ফেলার মধ্যে কাজ খতম। একটু বাদেই আমরা দেখব, এই কাজটা র‍্যাম করে। আবার, আপনি যুগান্তকারী যে প্রবন্ধটা লিখলেন সেটা পরবর্তী সহস্রাব্দেও যাতে হারিয়ে না-যায়, তার জন্যে আপনি একটা সিডিতে পুড়িয়ে সেটা ব্যাঙ্কের লকারে রেখে দিলেন, এবং ব্যাঙ্কের ম্যানেজারকে বলে দিলেন, যেন আগামী দু-এক হাজার বছর এটা তুলে রাখে। দু-এক হাজার বছর পরে সিডিটা যদি থাকে, তাতে আপনার তথ্যটাও তোলা রইল। বার করে কম্পিউটারে ভরা মাত্র আবার সেটা চটকানো যাবে। তবে ততদিনে আমাদের চেনা অবয়বের কম্পিউটার আর সিডি থাকবে মিউজিয়ামে, নালন্দা বিশ্ববিদ্যালয় থেকে পাওয়া জিনিষপত্র আমরা যেমন সাজিয়ে রাখি। তথ্য ধরে রাখাটা হয় নানা রকম। ধরুন, হিশেব করছেন, হিশেব চলাকালীন হিশেবের প্রতিটি সংখ্যাকে মনে রাখতে হচ্ছে মেশিনের, ভুলে গেলে তো হিশেবটাই আর হবেনা। এটা হল চকিত কটাক্ষের স্মৃতি কয়েক লহমার জন্যে ধরে রাখা, মানে র‍্যাম বা ভার্চুয়াল মেমরি, আসছি আমরা একটু বাদেই। আবার ধরুন, পরপর আপনি অনেকগুলো মাসের হিশেবের সমগ্রটা একসঙ্গে দেখছেন, তার মানে আগের মাসের তথ্যগুলোও মেশিনকে কোথাও তুলে রেখে দিতে হয়েছে — সচরাচর এই কাজে হার্ডডিস্ককে ব্যবহার করা হয়, তবে অন্য উপাদানও ব্যবহার করা যায়। মানে, নানা রকম তথ্য নানা ভাবে তুলে রাখতে হচ্ছে মেশিনকে তথ্য চটকানোর কাজটা করতে গিয়েই। এটাই হল ইনফর্মেশন হোল্ডিং বা তথ্য ধারণ।

৫।। কন্ট্রোল বা নিয়ন্ত্রণ। কোন অংশটা কোথায় কী করে চলেছে তার উপর একটা নিয়ন্ত্রণ তো থাকতে হবে। এইখান থেকে তথ্য পড়ো, এইখানে থামাও। এতটা অব্দি ঢোকাও। এতটা অব্দি এই ভাবে চটকাও। এইবারে এই অব্দি ফলাফলটা বার করে দাও এই ভাবে এবং এইখানে। এই কাজটাকে খেয়াল করুন, আলাদা করে খেয়াল করুন, কারণ কাজটার সঙ্গে অন্য কাজগুলোর একটা ব্যাপক প্রভেদ আছে। একটা তত্ত্বগত ঝাঁকুনি আছে। অন্য কাজগুলো আমাদের অভ্যস্ত ধাঁচের — কম্পিউটার নামক মেশিন এবং তথ্যের মধ্যেকার কাজ। আগের চারটে কাজই ভাবুন, যা ক্যালকুলেটরও করে। ক্যালকুলেটরের কিপ্যাড ইনপুট নেয়, ডিসপ্লে প্যানেলগুলো আউটপুট দেয়, মেমরি প্লাস বা মেমরি মাইনাস করে আমরা মেমরিতে রাখা সংখ্যাগুলো দেখি, মানে একটা মেমরি বা তথ্য ধারণও আছে, অফ করলেই যা উড়ে যায়। এবং তথ্য চটকানো তো আছেই, সেটাই তো কাজ ক্যালকুলেটরের। ধরুন আপনি ‘2’ আর ‘3’ যোগ করলে যে ‘5’ হয় এটা ক্যালকুলেটরের কাছ থেকে জানতে চাইছেন, তার মানে ‘2’ আর ‘3’ এই দুটো আঙ্কিক তথ্যকে চটকে ফলাফল ‘5’ বার করছে ক্যালকুলেটর। এটা করে ক্যালকুলেটরে বৈদ্যুতিন বা ইলেকট্রনিক সার্কিট অংশটা। কিন্তু, এই পাঁচ নম্বর মানে নিয়ন্ত্রণের কাজটা ক্যালকুলেটরের কোন অংশটা করছে? ভাবুন, খুব ভাবুন। পারছেন ভাবতে? কোন অংশটা? পারবেন না, কারণ, এই কাজটা ক্যালকুলেটর করছেই না। করছেন আপনি, আপনার মাথায় ভরা আছে, কী কী সুইচ টিপতে হবে, ‘2’, ‘+’, ‘3’ — এই অব্দি টেপার পর তথ্য দেওয়া আর তাদের কী করতে হবে সেই বলাটুকু শেষ হল, এবার টিপতে হবে ‘=’, মানে এবার আপনি হিশেব করতে বললেন। এখুনি হিশেব শেষ করতে না-বলে যদি আরো যোগ করতে চাইতেন তখন আপনাকে আবার টিপতে হত ‘+’, এবং তারপর আবার কোনো সংখ্যা। তার মানে নিয়ন্ত্রণটা কে করছে? আপনি। কী করবেন সেটা কোথায় রয়েছে? আপনার মাথায়। প্রোগ্রাম কাকে বলে? যা দিয়ে আপনার মাথার মধ্যে থাকা ওই নিয়ন্ত্রণের কাজটা কম্পিউটারের হাতে দিয়ে আপনি প্রাণের সুখে লেজ দোলাতে পারেন। বাই চান্স, লেজ যদি না থাকে, এই সমস্যাটা হতেই পারে, আমারো তো নেই বলেই জানি, মুন্ডু দোলান মৃদুমন্দ হাওয়ায়, একই আরাম পাবেন। কিন্তু কাজটার পার্থক্যটা খেয়াল করলেন কী? এটা একটা কাজ যা আর তথ্য আর মেশিনের মধ্যের দেওয়া নেওয়ার স্তরে নয়, এটা হল তথ্য আর মেশিনের মধ্যেকার কাজগুলোকে নিয়ন্ত্রণের কাজ। মানে টেকনিকাল চ্যাংড়ামো করে বলা যায়, মেটাকাজ, কাজের বিষয়ে কাজ। এই গোটাটা নিয়ে বিশদ করে আলোচনায় আসব আমরা এই পাঠমালার শেষ দিনে মানে দশ নম্বর দিনে।

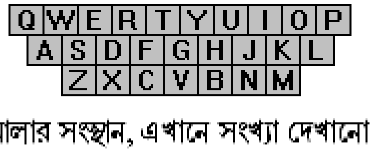
কম্পিউটারের যৌক্তিক গঠনের এই ছকটা মেনেই এবার বিভিন্ন ভৌত উপাদান বা ফিজিকাল ডিভাইসগুলো লাগানো এবং ব্যবহার করা হয়। সেই খুঁটিনাটিগুলো এবার একটু ধরা যাক।

০.২।। ইনপুট বা কম্পিউটারে তথ্য ঢোকা

ইনপুট বা তথ্য ঢোকানোর যে উপায়টা সবচেয়ে বেশি ব্যবহৃত হয় সেটা কিবোর্ড। এই কিবোর্ড জিনিষটা আমাদের অভ্যস্ত — প্রতিদিনের। কিন্তু আজ আমরা যে চেহারায় কিবোর্ডকে দেখি তার একটা স্বতন্ত্র ইতিহাস আছে। সেটা এক দিনে আসেনি। এবং কিবোর্ডের চেহারা শুধুমাত্র ওই একটাই নয়। আরো একাধিক রকমের হয়। আমরা যে কিবোর্ডে অভ্যস্ত তাকে বলে কোয়েরটি কিবোর্ড। এই নামটা এসেছে কিবোর্ডের নিচের যে ব্লকটা — মানে অক্ষরের ব্লক, তার একদম উপরের ছ-টা অক্ষরের নাম নিয়ে — ‘Q’, ‘W’, ‘E’, ‘R’, এবং ‘T’। নাম থেকে মিলিয়ে নিন।

কোয়েরটি কিবোর্ড।। কোয়েরটি বা QWERTY কিবোর্ড লেআউট হল আমাদের অভ্যস্ত কিবোর্ডের চাবির সজ্জা। এর বাইরে

কোয়েরটি (QWERTY) কিবোর্ড লে-আউট



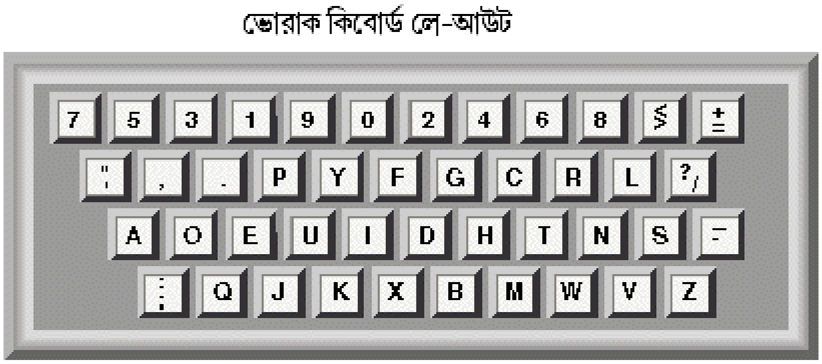
বর্ণমালার সংস্থান, এখানে সংখ্যা দেখানো নেই

অন্যরকম হয় — যেমন ভোরাক। এটা আমি বই থেকে জানি। আজ পর্যন্ত কখনো চোখে দেখিনি। তবে নেটে একটা কোনো

লেখায় পড়েছিলাম যে ওই ভোরাক বোধহয় আরো শরীর সংস্থানের বিজ্ঞান মোতাবেক তৈরি। কোয়েরটি নামটা এসেছে এই কিবোর্ডের বর্ণমালাটুকুর উপরে বাঁদিকের ছ-টা চাবি মিলিয়ে।

তবে কিবোর্ডে লে-আউট কেবলমাত্র এই একটাই নয়। শুনেছি যে, ভোরাক কিবোর্ডটা এই কোয়েরটির চেয়ে ব্যবহারবিধির বিজ্ঞান বা এর্গোনমিক্স অনুযায়ী অনেক বেশি সঠিক। এর্গোনমিক্স মানে মানুষের ব্যবহার্য বস্তুগুলোর আকার আকৃতি প্রকৃতি বিষয়ক বিজ্ঞান — যাতে করে মানুষ সবচেয়ে সুস্থ রকমে, সবচেয়ে কম ক্লান্তিতে সবচেয়ে বেশি কাজ করতে পারে। আমাদের চোখের সামনে আমরা সবচেয়ে নাটকীয় এর্গোনমিক বদল দেখেছি চেয়ারের আকৃতিতে। আজকাল চেয়ার তৈরিই হয় মেরুদণ্ড তথা শরীরের সুস্থতার কথা মাথায় রেখে — এই ডিজাইনগুলো এর্গোনমিক।

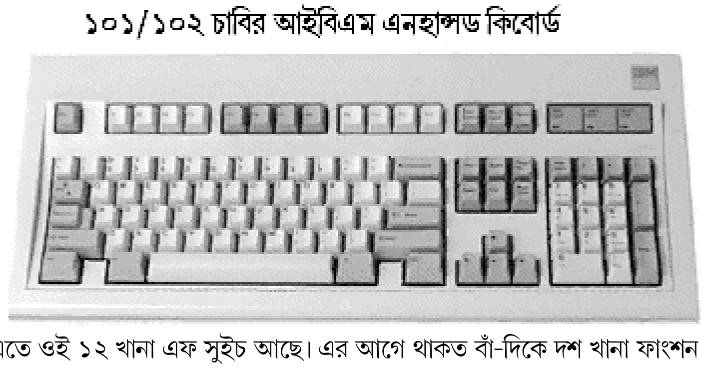
ভোরাক কিবোর্ড।। এই কিবোর্ড লে-আউট তৈরি করেন অগাস্ট ভোরাক এবং উইলিয়ম ডিলি, ১৯৩৬-এ, তখন ব্যাপকভাবে



জনপ্রিয় কোয়েরটি কিবোর্ডের বদলি হিশেবে — আরো দ্রুত যাতে টাইপ করা যায়। এতে চাবিগুলোও সেইভাবেই বসানো হয়েছিল। সবচেয়ে বেশি ব্যবহৃত অক্ষরগুলো যাতে সবচেয়ে সহজে, মানে সবচেয়ে কম আঙুল নাড়িয়ে পাওয়া যায়। যে অক্ষরগুলো প্রায়ই জোড়ায় জোড়ায় আসে তাদের পরস্পরের থেকে দূরে পাঠিয়ে দেওয়া হয়েছিল — যাতে আলাদা আলাদা হাতে তাদের পাওয়া যায়।

যে চেহারায় আজ আমরা কিবোর্ডকে দেখি তার মধ্যেও একটা দীর্ঘদিনের কম্পিউটার হার্ডওয়ার এবং সফটওয়ার বিবর্তনের ইতিহাস লুকিয়ে আছে। এখানে ছবিতে সেটা একটু দেখানোর চেষ্টা করলাম। এই ইতিহাসটা বেশ দীর্ঘ এবং আকর্ষনীয়। কিন্তু আমাদের এই পাঠমালার এক্তিয়ারের বাইরে। বাজারের এবং প্রযুক্তির বদলের সঙ্গে সঙ্গে মানুষের চিন্তাপ্রক্রিয়ার বদলের বেশ একটা চমৎকার গ্রাফ পাওয়া যায় তাতে।

আইবিএম এনহান্সড কিবোর্ড। ১০১/১০২ খানা চাবির। এই এনহান্সড কিবোর্ড এসে পুরোনো পিসি আর এটি কিবোর্ডকে



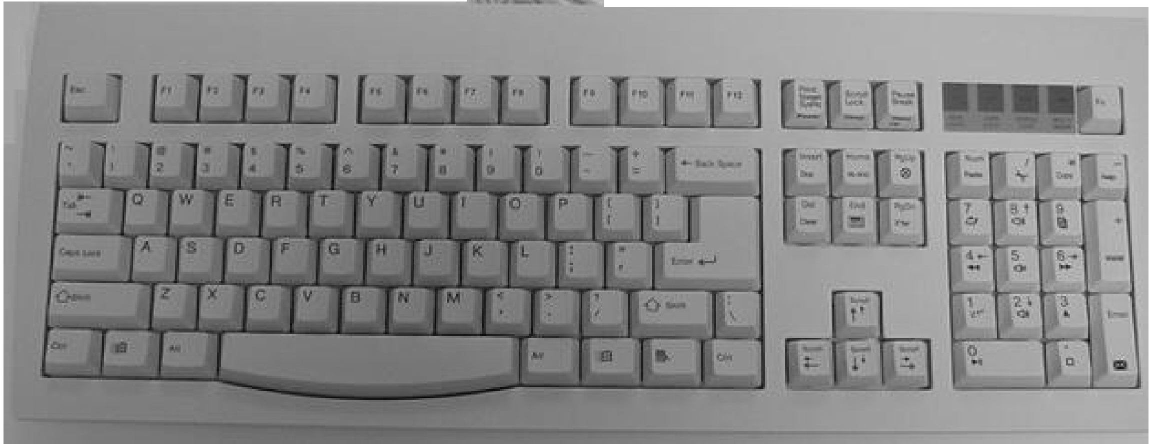
সরিয়ে দিয়েছিল। এতে ওই ১২ খানা এফ সুইচ আছে। এর আগে থাকত বাঁ-দিকে দশ খানা ফাংশন চাবি। এছাড়া এই এনহান্সড কিবোর্ডে নতুন এলো কনট্রোল (Ctrl) এবং অল্ট (Alt) চাবি। এবং এলো নিউমেরিক প্যাড এবং ওই অন্যান্য চাবি আর আলোগুলো। এই আইবিএম এনহান্সড এবং অ্যাপল এক্সটেন্ডেড কিবোর্ড দুটো প্রায় একই।

অ্যাপল এক্সটেনডেড কিবোর্ড।। ১০৫ খানা সুইচের। এই কিবোর্ড থেকেই অ্যাপল প্রথম উপরে এফ-সুইচের সিরিজ আমদানি করে পিসি কিবোর্ডে, এফ ১ থেকে এফ ১২, আইবিএম মেশিনে যা আগে থেকেই ছিল। পরে আমরা দেখব এই এফ সিরিজ

১০৬ চাবির অ্যাপল এক্সটেনডেড কিবোর্ড

ভীষণ বেশি ভাবে ব্যবহার হয় লিনাক্সে, ভৌতিক বা ভার্চুয়াল টার্মিনালের ব্যবহারে। এই এফ সিরিজ এবং কোন সুইচ কোথায় কতটা জায়গা নিয়ে থাকবে এই লে-আউট এবং নিউমেরিক প্যাডের, অল ক্যাপস সুইচের বা স্ক্রোল-লক সুইচের সঙ্গে আলো — এই গোটাটা মিলিয়ে — এই কিবোর্ড আর আইবিএম এনহান্সড কিবোর্ড বা তার যে প্রতিরূপগুলোর সঙ্গে আমরা সচরাচর

আমাদের খুব চেনা কিবোর্ড



অভ্যস্ত — এরা প্রায় একই। এদের ড্রাইভারগুলোও কাজ করে প্রায় একই রকমে। যা কিন্তু ভোরাক কিবোর্ডের বেলায় সত্যি না। ভোরাকটা ভয়ানকভাবে আলাদা। ছবিতে দেখুন।

তবে শুধু কিবোর্ড কেন, ইনপুট বা কম্পিউটারে তথ্য ঢোকার কাজটা এমনকি হার্ড ডিস্ক থেকেও হতে পারে। কোনো একটা ফাইলে হয়ত রয়েছে তথ্যটা। সেখান থেকে নিয়ে এলাম মেশিনের জ্যান্ত প্রক্রিয়ায়। তথ্যটা আসতে পারে ফ্লপি ডিস্ক থেকে। অন্য কোনো মেশিনে কাজ করা হয়েছিল — সেখান থেকে ফ্লপিতে করে নিয়ে আসা হয়েছে। বা আগের বার কাজ করার পর ফাইলটা ফ্লপিতেই সেভ করা হয়েছিল, রেখে দেওয়া হয়েছিল। তথ্য ঢোকানোর আর একটা উপায় সিডি। সিডিটাকে পুরে দিলাম সিডি-ড্রাইভে, সিডিতে আগে থেকেই পুড়িয়ে রাখা — মানে লেজার রশ্মি দিয়ে লেখা হয় বলে এর চালু নামটা হল পোড়ানো বা বার্নিং। তথ্য এই মেশিনে ঢুকতে পারে অন্য মেশিন থেকেও, নেটওয়ার্ক মারফত। নেটওয়ার্ক-টা হতে পারে ল্যান বা লোকাল এরিয়া নেটওয়ার্ক। হতে পারে ইন্টারনেট। কোনো একটা ওয়েবসাইটে গিয়ে একটা ওয়েবপেজে ক্লিক করলাম, পাতাটা আমার মেশিনে ঢুকে এল। এবার আমি ফাইলটাকে পড়তে পারি, তাকে রেখে দিতে পারি আমার ডিস্কে।

এই ফ্লপি বা হার্ড ডিস্ক বা সিডি আমার তথ্য রেখে দেওয়ারও মাধ্যম। যে ফাইল রেখে দিতে হবে তাকে সেভ করো ফ্লপি বা হার্ড ডিস্ক বা সিডিতে। শুধু রেখে দেওয়া কেন, আউটপুট বা তথ্য বার-করার মাধ্যমও হতে পারে এরা। এই যে জিএলটি ইশকুলের ক্লাসনোটসগুলো লিখে চলেছি, পর পর ফাইলে, ইশকুলের কেউ একটা চাইল, তাকে একটা ফ্লপি বা সিডিতে লিখে দিয়ে দিলাম ফাইলগুলো। সে আবার তার মেশিনে ওটা ঢুকিয়ে পড়ে ফেলতে পারবে।

আসলে আমরা কাজের ছকের ভিত্তিতে কম্পিউটারের অংশগুলোকে বুঝতে চাইছি। কাজের ভিত্তিতে যদি ভাগ করি, তাহলে কম্পিউটারের একটা কাজ হল তথ্য ঢোকানো। অনেক ভৌত উপাদানই আছে যা দিয়ে এটা করা যায়। কিন্তু তারা তখন প্রত্যেকেই কাজের নিরিখে এক — প্রত্যেকেই মেশিনে তথ্য ঢোকাচ্ছে। আবার ওই একই ভৌত উপাদানগুলো যখন অন্য কাজে লাগালাম, তথ্য তুলে রাখতে, কিম্বা বার-করতে, তখন তারা কাজের নিরিখে কম্পিউটারের অন্য রকমের একটা অংশ হয়ে গেল। তথ্য বার-করার, তথ্য তুলে-রাখার।

এবং একই ভৌত উপাদান নানা বিচিত্র ভাবেও ব্যবহার করা যেতে পারে। ধরুন যদি বলি, আপনার কম্পিউটারের হার্ড ডিস্ক — ওটার কাজ কী? সবাই একবাক্যে বলবে মূলত তথ্য তুলে রাখা, এবং প্রয়োজনমাফিক তথ্য ঢোকানো। কিন্তু, হুঁ হুঁ, আমি প্রবালকে দেখেছি, জিএলটি মধ্যমগ্রামের সভ্য, যে প্রায় পকেটে করে হার্ড ডিস্ক নিয়ে ঘোরে। এবং সঙ্গে তার কানেক্টর। আমি বলেছিলাম, তোর কাছে যে ডাউনলোড করা টেক্সটগুলো আছে, আমায় দিবি? বলল, ঠিক আছে, নেক্সট দিন তোমার জন্যে আমার বড় হার্ড ডিস্ক থেকে এই হার্ড ডিস্কটায় তুলে নিয়ে আসব।

এবং হার্ড ডিস্ক নিয়ে ঘোরা মানে বুঝতে পারছেন? আনতাবড়ি হিশেবে বলা যায় — ওর কুড়ি জিবির হার্ড ডিস্ক — মানে ধরুন মোটামুটি গোটা চল্লিশেক সিডি বা চোদ্দ হাজার ছশো তিরিশখানারও বেশি ফ্লপি। এতটা পরিমাণ তথ্য ফ্লপিতে নাড়াচাড়া করাটা একটা অসম্ভব ব্যাপার, তাই ফ্লপির কথা ছেড়েই দিন। সিডিতে করা যেত হয়ত, কিন্তু ওর কোনো সিডি বার্নার নেই, মানে সিডিতে লেখার ব্যবস্থা, বার্নার কেনার কোনো প্ল্যানও নেই — দিব্য চলে যাচ্ছে হার্ড ডিস্ক পকেটে ঘুরে। ওর নিজের মেশিনে কোনো নেট যোগাযোগও নেই, পাড়ায় সাইবার কাফেতে গিয়ে নেট করে। কথা বলে নিয়েছে, তাদের মেশিনে গিয়ে নিজের হার্ড ডিস্কটা লাগিয়ে নেয়, ব্রাউজ করে তুলে রাখে, পরে অপ্রয়োজনীয় অংশ ডিলিট করে, উড়িয়ে দেয়। প্রবাল আমায় যখন গল্পটা করছে আমি মনে মনে ভাবার চেষ্টা করছি সোদপুর এইচবি টাউনের সেই সাইবারকাফের মালিকের মুখটা — যখন তিনি প্রথমবার প্রস্তাবটা শুনলেন। এই গোটা প্রক্রিয়াটাই একান্তভাবে বিদঘুটে এবং বিপজ্জনক — এই প্রবাল-ব্র্যান্ড 'হাতে রইল হার্ড ডিস্ক' পদ্ধতি। হার্ড-ডিস্কটা যে কোনো মুহূর্তে নষ্ট হয়ে যেতে পারে, আর কোনো কম্পিউটারে গিয়ে তার ক্যাবিনেট খুলে সেটা লাগিয়ে নেওয়ার ঝঞ্ঝাট তো ছেড়েই দিন। এবং বেশির ভাগ লোক কম্পিউটারের ক্যাবিনেট খুলতে এত ভয় পায় যে আতঙ্কে হিম হয়ে যাবে এজাতীয় প্রস্তাবে। অর্থাৎ যে কোনো ভৌত উপাদানকেই, তার সম্ভাব্যতা অনুযায়ী, যে কোনো কাজের একক হিশেবে ব্যবহার করা যায়। করা হয় কি হয়না সেটা পরের প্রশ্ন — ভৌত উপাদানটার ভৌত ধর্মের প্রশ্ন।

মেশিনে তথ্য ঢোকানোর আর একটা বড় উপায় ইঁদুরের পেট টিপে কিঁচ কিঁচ মানে ক্লিক ক্লিক করতে থাকা। মাউস, বা মাউসের কোনো বিকল্প — মানে ট্র্যাকবল বা ওই জাতীয় কিছু। ধরুন আপনি এই যে ফাইলটা পড়ছেন, কোনো একটা ব্রাউজার মানে পরিদর্শক সফটওয়ার দিয়ে। যা দিয়ে আমরা ইন্টারনেটে বিভিন্ন ওয়েবপেজ চেখে বেড়াই, ইচ্ছে হলে নিজের মেশিনের ফাইল এবং ডিরেক্টরির সম্ভারগুলোও দেখতে পারি, সেই প্রায়োগিক সফটওয়ার বা অ্যাপ্লিকেশনকে বলে ব্রাউজার। এই ব্রাউজার দিয়ে আপনি সেই মেশিনের সেই ডিরেক্টরিটা খুললে ফাইলটা পেয়ে যাবেন, যেখানে ফাইলটা রাখা আছে। ব্রাউজার অনেক আছে, গ্নু-লিনাক্স হলে সচরাচর কংকোরার বা গ্যালিয়ন বা লিংক্স, আছে আরো অজস্র, আর উইনডোজ হলে ইন্টারনেট এক্সপ্লোরার বা নেটস্কেপ ন্যাভিগেটর। আমরা যখন তা দিয়ে কোনো মেশিনের কোনো একটা ডিরেক্টরি খুলি, সেখানে একটা বড় ছবি ফুটে ওঠে — তার মধ্যে পরপর সারি সারি ফাইলের ছবি — প্রত্যেকটার পাশে তার নিজের নাম। এই যে স্ক্রিনে এটা ফুটিয়ে তুলেছে কম্পিউটার — এটা কিন্তু তথ্য বার করা, বার করে সে স্ক্রিনের ছবিতে নিয়ে আসছে, তার নিজের উদরে অন্ত্র তন্ত্র কী আছে তার তালিকা, যাতে আপনি জানতে পারেন, তারপর ইচ্ছে বা প্রয়োজন মত তাদের নেড়ে চেড়ে দেখতে পারেন। ফাইলের ডিরেক্টরির তালিকা, মানে তথ্য। ছবিটাও তথ্য, অক্ষরে লেখা তালিকাটাও তথ্য। এবার সেই তালিকার মধ্যে যেই এই ফাইলের নাম লেখা ফাইলের ছবিতে আপনি আপনার মাউস নিয়ে গেলেন তখন কার্সর — মানে স্ক্রিনে মাউসের উপস্থিতির চিহ্নটা উত্থিত তর্জনীর মত বা ওইরকম কিছু হয়ে গেল। আপনি মাউসের বোতামটা টিপলেন : ক্লিক, ফাইলটা খুলে গেল। এই লেখাটা যদি আপনি নেট থেকে নামিয়ে থাকেন, তাহলে পাচ্ছেন একটা পিডিএফ ফাইলে। তাই, আপনি ফাইলটার উপর ক্লিক করা মাত্র, পিছনে, আপনার অগোচরে, মেশিন জাগিয়ে তুলল পিডিএফ ফাইল খোলার একটা সফটওয়ারকে, যে এবার ফাইলটাকে খুলে দেবে আপনার সামনে। পিডিএফ ফাইল পড়ার কোনো

একটা সফটওয়ার এই ফাইলের গোটা তথ্যটাকে, ছবি-তথ্যকে ছবি বানিয়ে, এবং অক্ষর-তথ্যকে অক্ষর বানিয়ে স্ক্রিনে ফুটিয়ে তুলল। বার করে আনল মেশিনে আবদ্ধ ফাইলের জঠর থেকে। এই ফাইলটাই খুলল কেন কম্পিউটার? কারণ, আপনি ক্লিক করেছিলেন এই ফাইলের ছবিতে। মানে এই ফাইলের ছবিতে ছবি-সুইচ নামক তথ্যের সঙ্গে সাঁটা ফাইল-নাম তথ্যটা কম্পিউটারকে পাঠিয়েছিলেন ফাইলটা খোলার নির্দেশ সহ। এই গোটাটাই তথ্য। যা আপনি পাঠালেন। মাউস ক্লিক করে।

মাউস ক্লিকটাও একটা খেলা, স্ক্রিনে ওই ছবির শরীরে কিছু কিছু জায়গা এমন করে তৈরি করা যেখানে মাউস নিলেই জায়গাটা অ্যাকটিভেটেড বা সক্রিয় হয়ে ওঠে, সেই সক্রিয়তা মানে, এবার মাউস-সুইচ ক্লিক করলেই ওই জায়গাটায় রাখা তথ্যটাকে মেশিনে পাঠিয়ে দেবে। মেশিন এবার এই ক্ষেত্রে দেখল যে সেটা একটা ফাইল নাম, এবং ওই ধরনের ফাইল কোন সফটওয়ার দিয়ে খুলবে সেটা মেশিনের আগে থেকেই জানা আছে, সেই সফটওয়ারকে জাগিয়ে তুলল মেশিন, সেই সফটওয়ার ওই তথ্যের মধ্যে পাওয়া নামের ফাইলটাকে খুলে আপনার সামনে হাজির করল। ধরুন বাজনা শোনার সফটওয়ার আলাদা, পিডিএফ ফাইল পড়ার সফটওয়ার আলাদা, এক ধরনের ফাইলকে অন্য ধরনের সফটওয়ার দিয়ে খুললে চলবেনা। মানে কম্পিউটার চলবে অবশ্যই, কিন্তু তাতে আপনার কাজ চলবে না। আপনি একটা ছবির ফাইলকেও টেক্সট বা লেখা হিশেবে খুলতে পারেন, স্ক্রিনে ফুটে উঠবে লাইন লাইন রাশি রাশি সংখ্যা চিহ্ন অক্ষর, তাতে আপনার ছবি দেখার নান্দনিক আগ্রহ কতটা পরিপূরিত হবে সেটা বলা কঠিন। আজকের দিনের শেষ দিকে আমরা এরকম একটা বিকট কাজ নিজেরাই করব।

মাউস।। একটা বহুল-ব্যবহৃত পয়েন্টিং ডিভাইস বা চিহ্নিত করার উপকরণ। প্রথম জনপ্রিয় হয় অ্যাপল কোম্পানির ম্যাকিনটোশ মেশিনের কল্যাণে। ক্রমশ যখন গুই বা গ্রাফিকাল ইউজার ইন্টারফেস, মানে কনসোলে টাইপ করে কমান্ড দেওয়ার বদলে ছবির মাধ্যমে কাজ করার পদ্ধতি, বাড়তে থাকে, ডস, উইনডোজ, অস/২ ইত্যাদি অপারেটিং সিস্টেমের সঙ্গেও ব্যবহার

হওয়া শুরু হয়। বিশেষত পার্সোনাল কম্পিউটার এবং ওয়ার্কস্টেশন মেশিনের দুনিয়ায়। মাউসের পুরো যন্ত্রপাতিটা ভরা থাকে একটা আবরণের মধ্যে যার তলার দিকটা সমতল, এবং উপর দিকটা হাতের তালু দিয়ে ধরার সুবিধের জন্যে তির্যক বর্তুল করে বানানো। উপরে দুটো বা তিনটে সুইচ, এবং এখন একটা চাকাও থাকে। এই সুইচ এবং চাকার যোগ থাকে নিচে একটা বলের সঙ্গে এবং সংযোগকারী তারের মাধ্যমে কম্পিউটারের সঙ্গে। মাউস নাড়ালে বলটা ঘোরে এবং তার সঙ্গে ঘোরে রোলার দুটো। ছবিতে দেখুন। রোলার দুটোর ঘোরা আসলে পরিমাপ করে কার্টেজিয়ান X অক্ষ এবং Y অক্ষ অনুযায়ী স্থান পরিবর্তনের হার, মানে মাউসটা অনুভূমিক এবং উল্লম্ব ভাবে কতটা নড়ছে — এর সঙ্গে তাল মিলিয়ে নড়ে পর্দার উপরে মাউসের স্থান-চিহ্ন মানে কার্সরটা। সচরাচর এটা হয় তীর-আকৃতির। এই তীরটা ছবির উপর দিয়ে নড়তে থাকাকালীন ছবির বিশেষ বিশেষ কিছু জায়গা,

সেইভাবে ব্যবস্থা করা থাকে, সক্রিয় হয়ে ওঠে। আমরা দেখি সুইচের মত ব্যবহার করছে ছবির সেই অংশগুলো। তখন আমরা মাউসের সুইচ টিপি। একেই বলে ক্লিক করা। মাউসে যখন একটা চাকা থাকে — সেই চাকাটা স্ক্রোলিং বা পাতা ওঠানো নামানোর কাজ করে — চাকার নড়াটাকে সচরাচর ডাকা হয় Z অ্যাক্সিস ম্যাপিং বলে।

পয়েন্টিং ডিভাইস বলতে আমরা বুঝি যা দিয়ে পয়েন্ট করে, চিহ্নিত করে স্ক্রিনের পটে আঁকা ছবির নানা নির্দিষ্ট জায়গাকে, সেই জায়গায় নিহিত আদেশ চালু হয়ে ওঠে চিহ্নিত করার পরে ক্লিক করলেই। শুধু মাউস বা ট্র্যাকবল ধরনের কোনো পয়েন্টিং ডিভাইস না, হতে পারে এই গোছেরই অন্য কিছু — ট্র্যাকপ্যাড, জয়স্টিক — এরকম অনেক কিছুই। জয়স্টিক লাগে মূলত গেমস খেলার জন্যে — আমাদের কলকাতা লাগ-এর গেমস-বিশেষজ্ঞ তথাগত এ বিষয়ে আপনাদের জানাতে পারবে — লাগে লিখবেন। অজস্র কোম্পানি অজস্র সিস্টেমে অজস্র রকমে এই কাজটা করার চেষ্টা করেছে — তার প্রমাণ পাবেন নেটে। যদি সেই পয়সা থাকে, একদিন নেটে গিয়ে মোট সম্ভাব্য পয়েন্টিং ডিভাইস গোনার চেষ্টা করতে পারেন, মায়াপ্রপঞ্চময় এই ব্রহ্মাণ্ডের স্বরূপ বুঝে যাবেন।

কম্পিউটারে তথ্য ঢোকার আরো আরো উপায় হতে পারে স্ক্যানার — বাস্তব ছবিকে কাগজ বা পট থেকে কম্পিউটারে ব্যবহারযোগ্য গ্রাফিক্স তথ্যের ফাইলে বদলে নিলেন, এবার কম্পিউটার স্ক্রিনেই ছবিটাকে নাড়াঘাঁটা করতে পারবেন — মানে ওই গ্রাফিক্স তথ্যটা মেশিনে ঢোকালো স্ক্যানার। হতে পারে ডিজিটাল ক্যামেরা — গ্রাফিক্স তথ্যকে ফিল্ম প্রিন্ট হওয়ার যন্ত্রে না-পাঠিয়ে পাঠাচ্ছে কম্পিউটারে। মাইক্রোফোন — আপনার অবিস্মরণীয় কণ্ঠস্বর থেকে আপনার পরের প্রজন্ম যাতে বঞ্চিত না-হয়, আপনি তাই সেটা অডিয়ো মানে শব্দসংক্রান্ত তথ্য করে পাঠিয়ে দিলেন মেশিনে এই মাইক্রোফোন দিয়ে। ভিডিও-ক্যাসেট রেকর্ডার এবং তার থেকে কম্পিউটার তথ্যে রূপান্তরিত করার কার্ড (কার্ড কাকে বলে এই জ্ঞান আপনি অচিরেই পাবেন) — সমভিব্যাহারে এরা দুজনেও তথ্য ইনপুট করে কম্পিউটারে। ভিডিওক্যাসেট থেকে কম্পিউটারে দেখনযোগ্য ফাইল বানায়।

০.৩।। আউটপুট বা কম্পিউটার থেকে তথ্য-বেরোনো

কম্পিউটার থেকে তথ্য বেরোনোর সর্বপ্রথম এবং সর্বপ্রধান উপায় হল স্ক্রিন মানে পিসির কনসোল। যা চাও তাই ফুটিয়ে তুলবে — অক্ষর বলো অক্ষর, ছবি বলো ছবি, আলোহীন অন্ধকার চাইলে তাও — যদি মেশিন বা কনসোল অফ করে দাও। আমার পারিবারিক অভিজ্ঞতা থেকে বলতে পারি, এই তৃতীয়টাও মাঝে মাঝে বেশ জনপ্রিয় প্রকার হয়ে ওঠে বৈকি।

তথ্য বার করার আর একটা জনপ্রিয় উপায় হল প্রিন্টার। কম্পিউটারের ইতিহাসের, এই পাঠমালার তিন আর চার নম্বর দিনে বিশদ ভাবে দেখব আমরা, একদম প্রথম দিকে এই তথ্য বার-করার প্রায় গোটাটাই ছিল কাগজে ছেপে। আজকের দুনিয়ায় যদিও, গোটা সাইবারতথ্যের একটা খুব ছোট ভগ্নাংশই শেষ অব্দি ছাপা হয়। এটা আমার বন্ধুদের অনেককেই আমি বোঝাতে পারিনা — স্ক্রিনে পড়া অভ্যেস করো। তাদের সব ঝিনচাক প্রিন্টার, আমার মত মজদুর-ব্র্যান্ড ডট-ম্যাট্রিক্স নয়, যা খুশি যখন খুশি ছেপে নেয়, দু চারটে আমাকেও দেয় বটে।

আইসি বা ইন্টিগ্রেটেড সার্কিট।। একটা আইসি বা ইন্টিগ্রেটেড-সার্কিট-এ থাকে বৈদ্যুতিন বা ইলেকট্রনিক যোগাযোগে পরস্পর যুক্ত মানে সার্কিটাবদ্ধ একাধিক ট্রানজিস্টর এবং রেজিস্টর — এই সার্কিটসহ এদের গোটাটাকে একটা বিশেষ প্রযুক্তি দিয়ে

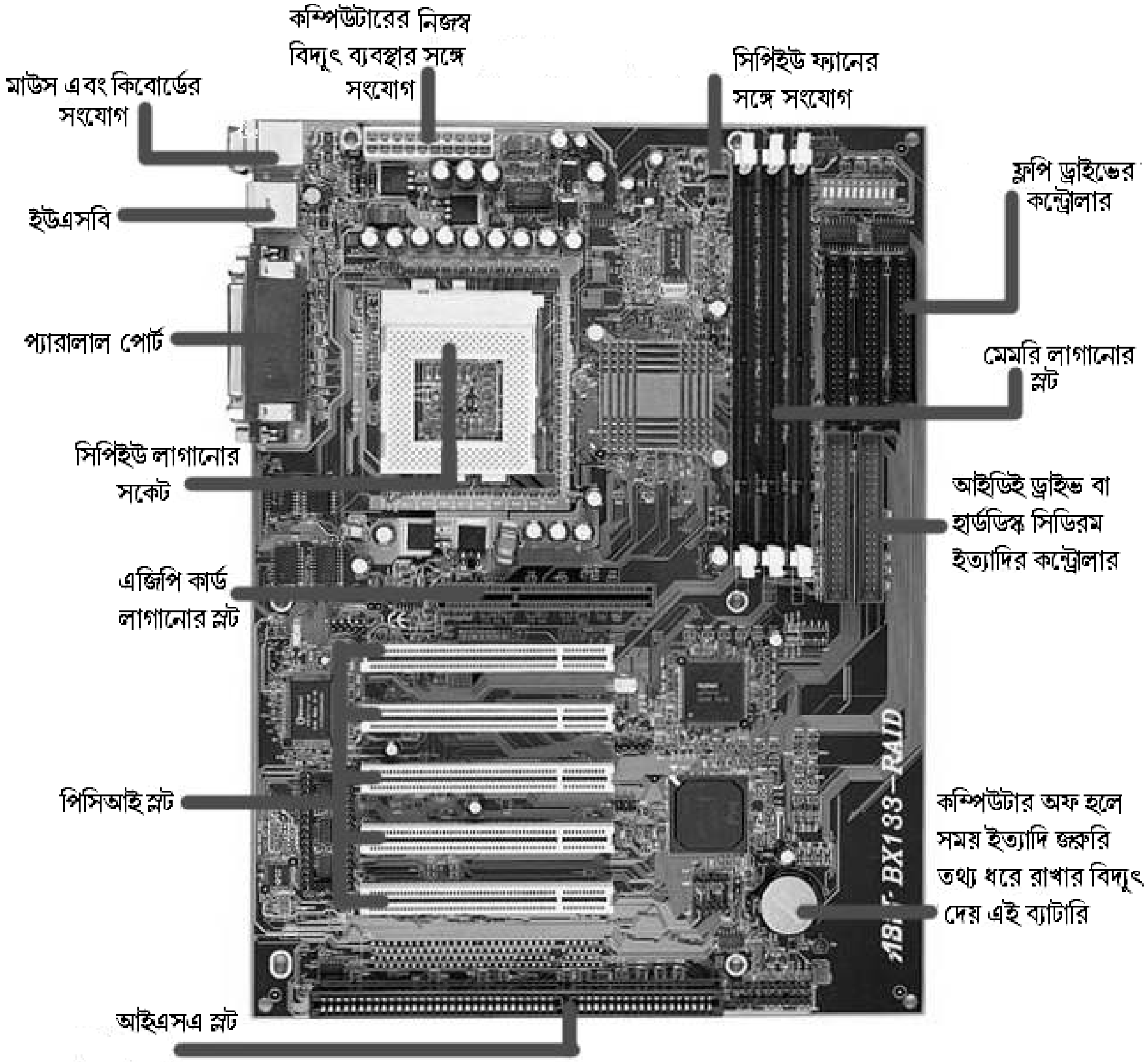
আইসি বা ইন্টিগ্রেটেড সার্কিট

দেগে দেওয়া হয়, গেঁথে দেওয়া হয় একটা ছোট্ট সিলিকন ক্রিস্টালের চিপে। এবং সচরাচর এই ইন্টিগ্রেটেড সার্কিটগুলোকে

একটা থেকে অন্যটাকে আলাদা করা হয় এদের মধ্যে কতগুলো ওই ট্রানজিস্টর বা রেজিস্টর ইত্যাদি উপাদান রইল — তার ভিত্তিতে। কম্পিউটারের সিপিইউ-ও এইরকম একটা একক চিপ। দশ লক্ষেরও বেশি ট্রানজিস্টর সম্পন্ন এই ইন্টিগ্রেটেড সার্কিটটাকে দেগে দেওয়া হয় একটা মোটামুটি ভাবে এক বর্গইঞ্চির সিলিকন চিপে। এখন যদিও একাধিক চিপের কম্পিউটারও হচ্ছে — এদের ডাকা হয় সিমেট্রিক-মাল্টি-প্রসেসিং মেশিন বলে, আমি নিজে এখনো শুধু এই এক চিপের মেশিনই চোখে দেখেছি। আমার পরিচিত পরিজনদেরও কারুর নেই। থাকলে দেখে আসতাম। এক চিপের সিপিইউ সবচেয়ে জনপ্রিয় যেগুলো — পেন্টিয়াম সিরিজ, অ্যাথলন, সাইরিক্স ইত্যাদি।

একটা জিনিষ খেয়াল করুন — কম্পিউটার তার প্রিন্টারে যা পাঠাচ্ছে সেই বারমুখো তথ্য — তা কিন্তু নিছক অক্ষর না, আরো বহু বহু কিছু। ধরুন এই লেখাটার এই বাক্যটাকেই। এটা পড়ছেন ১৪ পয়েন্ট অক্ষরে, ১১ পয়েন্টে নয়, ৯ পয়েন্টে নয়, **১৮-ও নয়,** সরল সাদামাঠা ১৪ পয়েন্টে। লেখাটার একদম প্রথম পাতার ব্রহ্মতালুতে দেখুন ‘গ্নু-লিনাক্স ইশকুল’ বাঁদিক ঘেঁষে, তার পরের প্যারাটাই ডানদিক, তার পরেরটা একদম মাঝামাঝি। এইসব কায়দাবাজি ছাড়াও ছবি আছে অনেক। অক্ষর ছাড়াও এই সমস্ত তথ্যই কম্পিউটার পাঠাচ্ছে তার স্ক্রিনে বা প্রিন্টারকে। কী ছাপবে, কী করে ছাপবে, তার সব ডিটেইলস। এই ফাইলটাকে যদি নিছক টেক্সট আকারে খোলেন, তারও উপায় আছে, দেখবেন। একটা লেখার ফাইলের এই গোটা তথ্যটার খুব সামান্য অংশই অক্ষর।

মাদারবোর্ড।। কম্পিউটারের হৃদপিণ্ড। মূল সার্কিট বোর্ড — যার গায়ে লাগানো থাকে কম্পিউটারের জরুরি সমস্ত উপাদান।

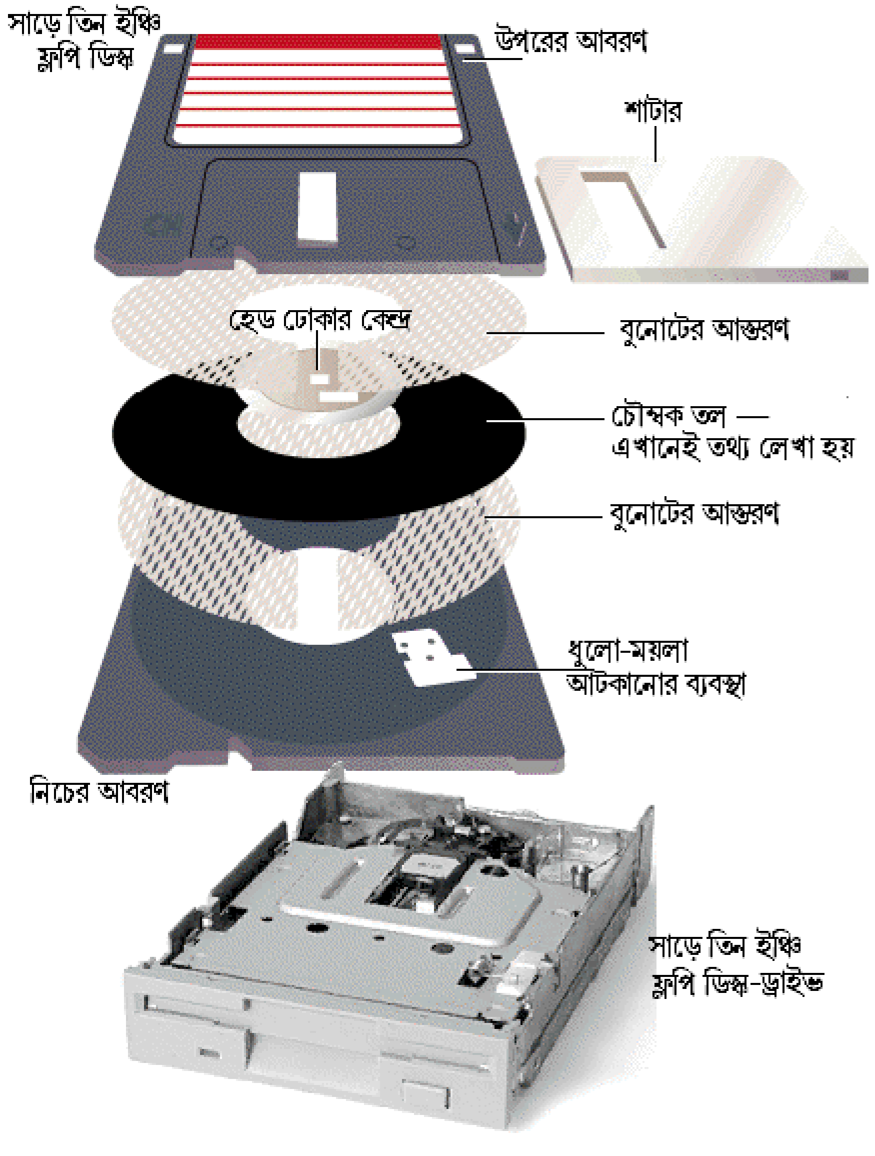


এই বোর্ডেই বসবাস করে সিপিইউ, মূল মেমরি, এদের তথ্য চলাচলের সার্কিট সংযোগ, বাস কন্ট্রোলার (সেটা কী বস্তু তা

বুঝতে আমাদের কিঞ্চিত বিলম্ব হবে, পাঠমালার দু নম্বর দিনের শেষ অব্দি) এবং কানেক্টরগুলো — যারা এই বিভিন্ন উপাদানগুলোর মধ্যে বৈদ্যুতিক এবং তথ্যের সংযোগ ঘটায়। অন্য আর সমস্ত বোর্ড, মানে এক্সটেন্ডার বোর্ড, অ্যাডাপ্টার কার্ড, পরিবর্দ্ধিত মেমরি, ভিডিওকার্ড ইত্যাদি অন্যান্য ইনপুট-আউটপুট কার্ড — এ সমস্তই লাগানো হয় এই মাদারবোর্ডের গায়ে। বাস কানেক্টর এদের মধ্যে তথ্য সংযোগ ঘটায়। এখানে আমরা একটা মাদারবোর্ডের ছবি দিলাম, এটা নেট থেকে পাওয়া, কারণ, অনেক চেষ্টা করেও কিছুতেই একটা ডিজিটাল ক্যামেরা যোগাড় করতে পারিনি, যা দিয়ে এই লেখা যে মেশিনে চলছে সেই মেশিনের অংশগুলোকে দেখানোর প্ল্যানটা কাজে আনা যেত। শুধু মধ্যে বাংলা লিখে এর কপিরাইটের সঙ্গে খুনশুটি করা হল। এখানে দেখুন অনেকগুলো অংশ আপনি এই মুহূর্তে বুঝতে পারবেন না অনেকেই। গোটা পাঠমালাটা পড়া হতে আর একটু পারবেন, তা নিয়ে চিন্তিত হবেন না। নিজের মেশিন একবার খুলে দেখে ফেলুন, সেটা বেশ উপকার করে। পিউ এবং অশেষ এই দুজনের সঙ্গেই এভাবে শুরু করেছিলাম, মেশিনটা খুলে ফ্যাল, দ্যাখ, কী কী চিনতে বা বুঝতে পারছিস। যদি গন্ডগোল হয়, হার্ডওয়ারের লোক ডাকতে হবে, সে তো এমনিতেও মাঝে সাঝেই হয়। তাকে আসল গল্পটা বলতেই হবে, এর কী মানে আছে? প্রকৃত বিশ্বাসের সঙ্গে বিশ্বকর্মাকে একবার স্মরণ করে নিলে আর পাপ হবেনা।

এই ফাইলটা লেখা হচ্ছে আমার পিসিতে। আমার মেশিন সেটা মোডেম মারফত নেটে পাঠাচ্ছে — মানে আউটপুট-টা হচ্ছে অন্য কম্পিউটারে। তারপর সেই কম্পিউটার থেকে আবার নেট মারফত ইনপুট হচ্ছে আপনার পিসিতে। অথবা প্রথমে ফ্লপি বা সিডিতে বার করা হচ্ছে — সেই আউটপুটটা ফ্লপি বা সিডি মারফত পৌঁছে যাচ্ছে আপনার মেশিনে। এই নেট বা ফ্লপি বা সিডিটা, বা প্রবালের হার্ডডিস্কও — এখানে তথ্য বার করার উপায়।

ফ্লপি ডিস্ক এবং ফ্লপি ডিস্ক ড্রাইভ।। তথ্য ঢোকানো এবং বার করার একটা জনপ্রিয় কিন্তু অনিশ্চিত উপায়। বাঁদিকে একটা ফ্লপি



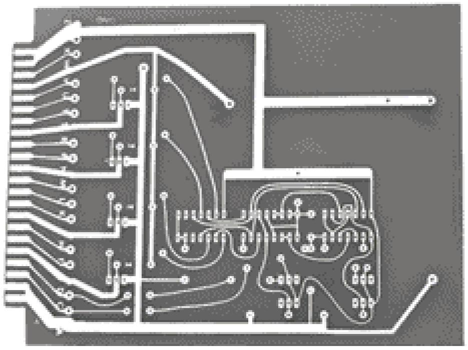
ডিস্কের ভিতরের উপাদানগুলো দেখানো হল। ফ্লপি হল তথ্য ঢোকানো এবং বার-করার, ইনপুট-আউটপুটের একটা বিদ্যুত-চুম্বকীয় মাধ্যম। একটা মোটর ফ্লপি ডিস্কটাকে ঘোরায়। আর একটা মোটর এক বা একাধিক রিড-রাইট হেড বা তথ্য লেখা-পড়ার ছুঁচকে ওই তথ্য লেখার চৌম্বক আস্তরণের তলের উপর দিয়ে নড়ায়। এই তথ্য যাতায়াত করে কম্পিউটার থেকে ফ্লপিতে যখন ফাইল লেখা হচ্ছে, আর ফ্লপি থেকে কম্পিউটারে, যখন ফাইল পড়া হচ্ছে। আগে ৫.২৫ ইঞ্চির ফ্লপিও হত, আমরা এখানে ৩.৫ ইঞ্চির ফ্লপি দেখিয়েছি। এত খারাপ হয় যে ফ্লপি ব্যবহার করার প্ল্যান থাকলে কোনো ফ্লপি-কোম্পানির কর্মচারীর সঙ্গে হটলাইন বানানোর তাল করা উচিত।

গান না-চালিয়ে লিখতে বা পড়তে পারিনা — সেই গানটা চলছে সাউন্ডবক্সে — অডিয়ো তথ্য বেরিয়ে আসছে। এই দিলীপ রায়ের গানগুলো ক্যাসেট থেকে আমার জন্যে এমপিথ্রি মানে কম্পিউটার-ব্যবহারযোগ্য ফাইল বানিয়ে দিয়েছে অশোকদা। সেখানে ইনপুট হয়েছিল বাস্তব কাঁচা র অডিও, অশোকদা সেটাকে টিমিডিটি বলে একটা সফটওয়ার দিয়ে এমপিথ্রি করে সেই ফাইল সিডিতে করে বার করে এনে আমায় দিয়েছিল। আমার মেশিনে ঢুকে সে আবার সাউন্ডবক্স দিয়ে আউটপুট হচ্ছে। ওই সিডিটা ছিল রিরাইটেবল, বারবার একই জায়গায় পুরোনো তথ্য উড়িয়ে নতুন তথ্য পুড়িয়ে দেওয়া যায়। আউটপুট ডিভাইস।

একটা জিনিষ ফের একবার মনে রাখুন, এই বার-হওয়া তথ্য বা আউটপুটটা কিন্তু গোটাটাই হল পরপর অগণ্য ০ এবং ১-এর একটা প্রবাহ, বাস্তব দিলীপ রায়ের বাস্তব গলার দার্ঢ্যটাও কিছু সবিশেষ ০ এবং সবিশেষ ১, যা ধরুন সুনিধি চৌহানের গলার তীব্র তীক্ষ্ণতার নিজস্ব ০ আর ১-এর প্রবাহের বিশেষত্ব থেকে আলাদা। অশোকদার ব্যবহৃত টিমিডিটি নামক সফটওয়ার ওই শব্দতরঙ্গের ওঠানামার বাস্তবতাটাকে বদলে দেয় সারি সারি শব্দরূপান্তরযোগ্য ০ এবং ১-এর বাস্তবতায়। আমরা পরে ফের আসব এটায়। দিন এক, দুই এবং পাঁচ-এ, বিশেষ করে।

সার্কিট বোর্ড।। এপক্সি বা ফেনলিক রেজিন জাতীয় কোনো তড়িৎ-অপরিবাহী জিনিষের চৌকো সমতল টুকরো — যার উপরে ইলেকট্রনিক ছোটছোট উপাদানগুলো লাগিয়ে দেওয়া হয়, বৈদ্যুতিক যোগাযোগ সহ। এখনকার প্রিন্টেড সার্কিট বোর্ডে তামার

সার্কিট বোর্ড



পাতে তৈরি সংযোগগুলো আগে থেকেই ছাপানো থাকে — ফোটোলিথোগ্রাফি করে। বোর্ডের একদিকে বা দুদিকেই থাকতে পারে বৈদ্যুতিন উপাদানগুলো। খুব জটিল সার্কিট বোর্ডে এই উপাদান এবং যোগাযোগগুলো তৈরি করা থাকে একাধিক স্তরে।

০.৪।। ইনফর্মেশন প্রসেসিং বা তথ্য চটকানো

এই তথ্য চটকানোর প্রায় গোটা কাজটাই করে আলু (ALU — **A**rithmetic-and-**L**ogical-**U**nit)। আলু ফলে থাকে সিপিইউ (CPU — **C**entral-**P**rocessing-**U**nit) — মানে আমরা যাকে প্রসেসর বলে উল্লেখ করেছি আমাদের ছবিতে — তারই একটা অংশে। আলু যোগ বিয়োগ গুণ ও ভাগ করে, এবং দুটো সংখ্যাকে তার সামনে ধরে দিলে, সঙ্গে সঙ্গে বলে দিতে পারে — এর মধ্যে কোনটা বড় — আলুর কী আশ্চর্য বুদ্ধি বলুন তো। এক কথায়, কম্পিউটারের সমস্ত লজিকাল অপারেশন বা যৌক্তিক হিশেব করে এই আলু। গ্নু-লিনাক্স ইশকুলের এই পাঠ পড়তে পড়তে আমরা প্রসেসর বিষয়ে আরো অনেক কিছুই জানতে পারব।

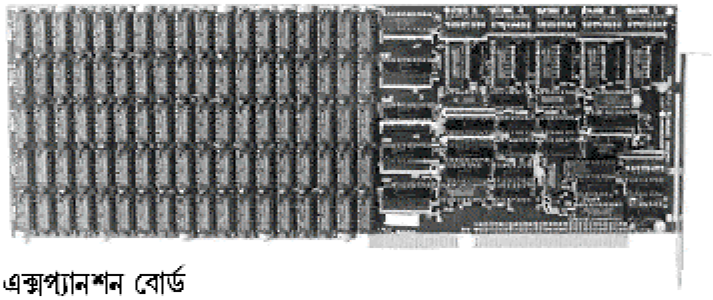
শুধু একটা জায়গা খেয়াল করুন — এটাকে সেন্ট্রাল প্রসেসিং ইউনিট বলে ডাকছি। তার মানে, কিছু অসেন্ট্রাল চিপও আছে নিশ্চয়ই। একটা কম্পিউটারের মাদারবোর্ডে (মানে এই চিপ মেমরি সার্কিট — এই সবকিছুর থাকা এবং খেলা

করার জন্যে একটা মায়ের কোল) থাকে একগুচ্ছ চিপ, কম করে গোটাপাঁচেক তো বটেই। তার এক একটার এক এক জাতের কাজ। কিন্তু তারা হল নামহীন নির্বাক কর্মচারীর মত, তাজমহলকে লোকে শাজাহানের নামেই চেনে, আপনার মেশিনকে আপনি ডাকেন পেন্টিয়াম ওয়ান টু থ্রি বা ফোর বলে, অ্যাথলন বা ডিউরন বলে, আগে ডাকতেন ৪৮৬ বা ৩৮৬ বলে — সেটা হল ওই কেন্দ্রীয় বা সেন্ট্রাল প্রসেসরের নামে। কারণ, এই প্রসেসরটা কী তার উপরেই মূলত নির্ভর করে অন্য চিপগুলো কী হতে পারে। শুধু অন্য চিপগুলোই নয়, অন্যান্য অংশগুলোও কী বা কী কী হতে পারে — তাও প্রায় ঠিক হয়ে যায় এই সিপিইউ ঠিক হয়ে যাওয়া মাত্রই। এই অন্যান্য চিপগুলোর বিষয়ে উল্লেখের বাইরে খুব বেশি কিছু আলোচনা আমাদের এই গ্নু-লিনাক্স ইশকুলের পাঠমালার এক্তিয়ারের বাইরে থাকবে।

০.৫।। ইনফর্মেশন হোল্ডিং বা তথ্য ধরে রাখা

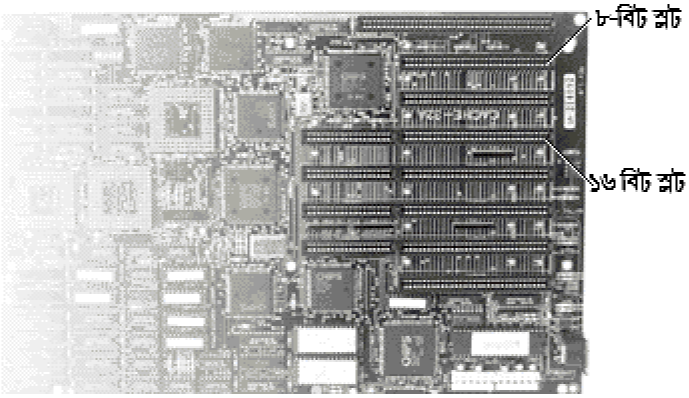
দীর্ঘ সময়ের জন্যেই হোক, বা অত্যন্ত সংক্ষিপ্ত, কম্পিউটারকে তথ্য ধরে রাখতেই হবে। পরিমাণটা খুব সামান্যও হতে পারে, আবার ডেটাবেস সার্ভারের মত বীভৎস বিপুলায়তনও হতে পারে। ধরুন যে তথ্যটুকুকে সে প্রসেস করছে, আমাদের ভাষায় চটকাচ্ছে, চটকানোর সময়টুকু অন্তত তথ্যটা তার হাতের তালুর মধ্যে থাকতে হবে — নইলে চটকাবে কী করে? এবং প্রতিটি চটকানোই অন্য প্রত্যেকটার থেকে আলাদা। কখনো ময়দাটাকে চটকাতে হবে ময়ান দিয়ে গরম জল দিয়ে, পরোটা বা লুচি হবে, কখনো ময়দা চটকাতে হবে টক দই দিয়ে, যদি ছোলে-বাটুরে বা রাধাবল্লভী হয়, আবার নানখাটাই হলে চটকাতে হবে চিনি আর সাদা তেল দিয়ে। আর কম্পিউটারের ময়দা তো শুধু একটা মাত্র রকমের নয়, হরেক আলাদা আলাদা কাজের আলাদা আলাদা সফটওয়ারের চাই আলাদা আলাদা ডো। চটকানোর জন্যে প্রয়োজনীয় সময়টুকু ধরে এই গোটা পদ্ধতি, উপকরণ — গোটাটাই ভরা থাকা চাই কম্পিউটারের মুন্ডুর ভিতর। (এই ডো ভরে রাখার জন্যে কম্পিউটারের নানা ধরনের মুন্ডু আছে। সেই আলোচনায় আমরা আসছি ঠিক এক প্যারাগ্রাফ বাদেই।)

এক্সটেন্ডার বোর্ড ।। সেই সমস্ত সার্কিট বোর্ড যাদের গুঁজে দেওয়া হয় মাদারবোর্ডে, কম্পিউটারের বাসে সংযুক্ত হয়ে এরা সেই কাজ করার সুযোগ দেয় শুধু মাদারবোর্ডে যা করা যেত না। এই এক্সটেন্ডার বা এক্সপ্যানশন বোর্ড দিয়ে মেমরি যোগ করা হয়,

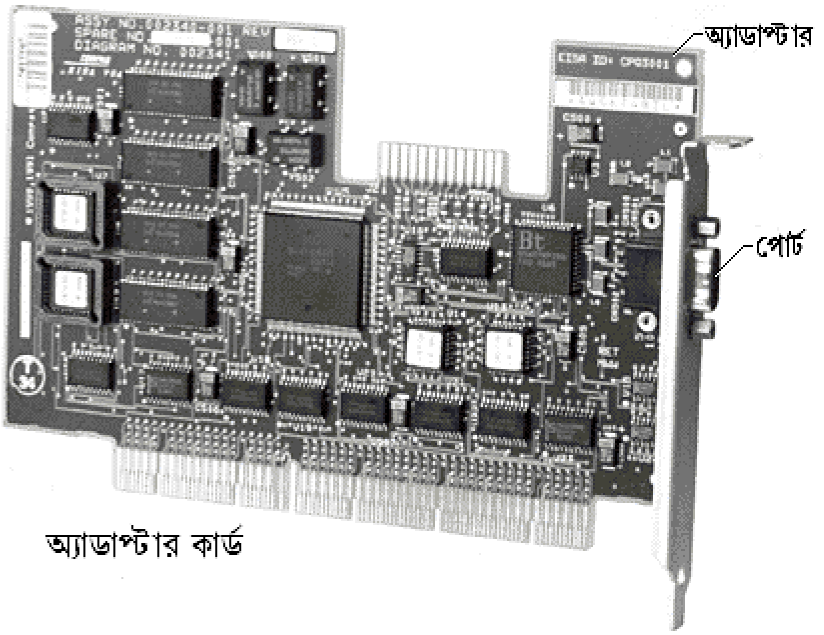


এক্সপ্যানশন বোর্ড

এক্সপ্যানশন স্লট (মাদারবোর্ডের শরীরে)



ডিস্ক ড্রাইভ কন্ট্রোলার লাগানো হয়, ভিডিও সাপোর্টকে উন্নততর করা হয়, প্যারালাল এবং সিরিয়াল পোর্ট যোগ করা হয়, এবং ইন্টারনাল মোডেম লাগানো হয়। মাদারবোর্ডে এক্সপ্যানশন বোর্ড লাগানোর জন্যে স্লট বা খাঁজ বানানো থাকে — ৮ বা ১৬ বিট স্লট (পরে বুঝব) এদের বলে এক্সপ্যানশন স্লট। এখানে যে এক্সপ্যানশন স্লট বা অ্যাডাপ্টার কার্ড আমরা দেখালাম, সেগুলো সবই বেশ পুরোনো আমলের, তার কারণটা তো আগেই বলেছি, কিন্তু তাতে খুব একটা কিছু এসে যাবে বলে মনে হয়না। মূল গঠনের বিশেষত্বটাই খেয়াল করার। এর সঙ্গে নিজের মেশিন খুলে মিলিয়ে নিলেই গোটা ব্যাপারটা স্পষ্ট হবে বলে মনে হয়।



অ্যাডাপ্টার কার্ড।। অ্যাডাপ্টার বলতে বোঝায় একটা প্রিন্টেড সার্কিট বোর্ড যা দিয়ে একটা কম্পিউটারে সেইসব পেরিফেরাল উপাদানকে লাগানো হয়, যেমন সিডিরম ড্রাইভ, মোডেম, জয়স্টিক ইত্যাদি — যেগুলোর জন্যে সরাসরি তার দিয়ে বা পোর্ট দিয়ে বা প্রিন্টেড বোর্ড দিয়ে কোনো সংযোগ প্রাথমিক ভাবে কম্পিউটারে করা থাকেনা। একটা অ্যাডাপ্টার কার্ডে একের বেশি অ্যাডাপ্টার লাগানো থাকতে পারে।

অর্থাৎ দু-ধরণের ইনফর্মেশনকেনাড়াচড়া করতে হয় আপনার মেশিনের। একটা হল কী চটকাব। অন্যটা হল কী ভাবে চটকাব। একটাকে বলা যায় ডেটা বা তথ্য। অন্যটা হল ইন্সট্রাকশন বা কমান্ড বা আদেশ। এই তথ্য এবং নির্দেশ দুটোই কিন্তু তথ্য — এই অর্থে যে মেশিনের কাছে দুটোই হল শূন্য আর এক-এর সমাহার। অজস্র অগণ্য ০ আর ১। তাই একই ধরনের তথ্য-ধারণ উপাদান দুটোকেই ধরে রাখতে পারে। একই স্মৃতি-কোষ বা মেমরি একই ভাবে নাড়াচাড়া করে দুটোকেই। হার্ড ডিস্কে দুটো ধরনের ফাইলই একই সঙ্গে থাকে। কিছু ফাইল প্রোগ্রাম ফাইল বা বাইনারি ফাইল বা এক্সিকিউটেবল ফাইল — মানে প্রোগ্রাম — চালানো হলে যারা চলে। আর অন্য ধরনের ফাইল হল তারা যাদের উপর এরা চলে। ধরুন একটা ওয়ার্ড প্রসেসর — মানে যা দিয়ে আপনি লেখেন। সেটা এমএস-ওয়ার্ড হোক বা ওপন-অফিস-অর্গ বা লোটাস স্মার্টস্যুট রাইটার — এরা প্রত্যেকেই এক একটা প্রোগ্রাম ফাইল। যারা চলে। আর আপনার লেখাটা যে ফাইলে রয়েছে সেটা হল সেই ফাইল যাদের উপর এরা চলে। প্রোগ্রাম ফাইলগুলো হল ইন্সট্রাকশন — কী ভাবে চটকানো হবে। আর ওই ডেটা বা তথ্য ভরা লেখার ফাইলটা হল তাই যাকে চটকানো হয়। এবার যাওয়া যাক ওই বিভিন্ন ধরনের তথ্য ধারণের মুন্ডুর আলোচনায়। পরপর তিনটে ছোট ছোট সাবসেকশনে।

তিন ধরনের স্মৃতিধারণোক্ষম মুন্ডুর প্রথমটাকে বলা যায় ক্ষণস্থায়ী মুন্ডু — ঠিক ততটুকু সময়ই ধরে রাখে যতটুকু সময় ধরে তথ্য এবং আদেশ চটকায়। যে মুহূর্তে চটকানো শেষ হয় এই স্মৃতিটা উবে যায়। এর নাম র‍্যাম (RAM — **R**andom-**A**ccess-**M**emory)। অন্য মুন্ডুটায় স্মৃতি হল স্থায়ী। গোটা মুন্ডুটাকে না-বদলে স্মৃতির কোনো অংশকে বদলানো সম্ভব না। একে বলে রম (ROM — **R**ead-**O**nly-**M**emory)। অন্য আর এক ধরনের মুন্ডু হল আপনার

ইচ্ছানির্ভর। যেমন আপনার মেশিনের হার্ডডিস্ক। যখন আপনার নিজের কম্পিউটারের হার্ডডিস্কটাকে মেদুর স্মৃতিভারাতুর বলে হবে তখন আপনি ঘচাঘচ করে ফাইল ওড়াতে থাকবেন। এবং, এটা অবশ্যম্ভাবী, ফাইল ওড়ানো মাত্রই আপনার মনে হবে — এমা, কী করলাম, এটা তো সাঙ্ঘাতিক কাজের ফাইল। চন্দ্রিল বলেছিল ফিল্ম বানানোর একটা ল — দি ল অফ দি লিস্ট পসিবল ক্যালামিটি। যে কেলোটা ঘটার মিনিমাম চান্সও নেই সেটাও ঠিক ঘটবেই শুটিং-এর দিন সকালে, ঠিক সেইরকম, সবচেয়ে প্রয়োজনীয় ফাইলটাকে আপনি একবার না একবার ভুল করে ওড়াবেনই ওড়াবেন। এই জন্যেই ব্রহ্মবৈবর্তপুরাণে বারম্বার লিপিবদ্ধ আছে নিয়মিত হার্ডডিস্ক ব্যাকআপের প্রয়োজনীয়তার কথা। যাকগে শাস্ত্রে যখন আছে আমরা আর বলে কী করব?

০.৫.১।। অনুদ্বায়ী বা স্থায়ী স্মৃতি — রম

আগেই বললাম, এই স্মৃতিটা কাজ শেষ হওয়ামাত্রই কর্পূরের মত উবে যায়না। একটু আগে আমরা কম্পিউটারের সেন্ট্রাল প্রসেসিং ইউনিটের মূল চিপ এবং অন্যান্য আরো যে চিপগুলোর কথা বলছিলাম — এরা এই ধরনের স্মৃতির একটা আধার। যেমন ধরুন বায়োস (BIOS — **B**asic-**I**nput/**O**utput-**S**ystem)। এই পাঠমালার পাঁচ নম্বর দিনে, এবং পরেও, আমরা একটা কম্পিউটার চালু হয়ে ওঠার গোটা প্রক্রিয়াটা অনেক খুঁটিয়ে আলোচনা করেছি — যার অন্য নাম বুট। এখন জাস্ট এইটুকু জানুন যে কম্পিউটারে যখন প্রাণ-প্রতিষ্ঠা হয়, বিদ্যুতযোগাযোগ চালু করার পর যখন ধীরে ধীরে স্ক্রিনে ফুটে ওঠে ছবি বা লেখা, কম্পিউটার ক্রমে আপনার আদেশ নেওয়ার এবং সেই আদেশ অনুযায়ী কাজ করার জায়গায় পৌঁছয় — সেই সময়টা জুড়ে অজস্র কিছু ঘটতে থাকে মেশিনের ভিতর। যার কিছুই প্রায় বাইরে থেকে আন্দাজ করা যায়না।

কিন্তু সব স্মৃতিই যদি কারেন্ট অফ করে দেওয়া মাত্র উবে যেত তাহলে প্রত্যেকবার নতুন করে কম্পিউটারকে ইনস্টল করতে হত, সমস্ত কাজের প্রোগ্রাম নতুন করে ভরতে হত কম্পিউটারে, বা কম্পিউটার কখনো অফ করা যেত না। তা যখন হয়না, তখন নিশ্চিত ভাবেই বলা যায় কিছু কিছু জিনিষ রয়ে যায় কম্পিউটারে। হার্ডডিস্কের ফাইলগুলো তো রয়ে যায়ই, মাদারবোর্ডের রমচিপগুলোয় রয়ে যায় আরো কিছু তথ্য বা আদেশ যারা কম্পিউটারকে অন হওয়ার পর থেকে কাজ করার জায়গা অব্দি পৌঁছে দেয় — হার্ডডিস্কের ফাইলগুলো নাড়াচাড়া করার জায়গা অব্দি। এই অব্দি কম্পিউটার ঠিক কী কী করবে, কোন পথে এগোবে সেটা ভরা থাকে এই এই বায়োস-এ। এই বায়োসটা থাকে মাদারবোর্ডের একটা চিপে — যার নাম বায়োস চিপ। সেন্ট্রাল প্রসেসিং ইউনিটের পাশাপাশি অন্যান্য অপ্রধান চিপগুলোর একটা। আর সিপিইউতে দেখানো থাকে এই বায়োস-রম-চিপের ঠিকানা। যাতে বুট করার সময় কম্পিউটার এই বায়োস চিপ থেকে পড়া শুরু করতে পারে।

এই বায়োস রম চিপে সেইটুকু ন্যূনতম নির্দেশ ভরা থাকে যা বিদ্যুত না-থাকা-কালীনও কম্পিউটার থেকে মুছে যাবেনা, যা থেকে কম্পিউটার জানতে পারবে বুটকালীন তাকে কী কী করতে হবে। এছাড়া অন্য আর যেসব রম থাকে কম্পিউটারে — তার মধ্যে পড়ে — প্রথমেই কোন কোন হার্ডওয়ার পার্টসকে, ভৌত যন্ত্রাংশকে সক্রিয় করে তুলতে হবে তার তালিকা। এই ধরনের রম চিপের কোনোটায় ভরা থাকে নির্মাতা এবং মডেলের খুঁটিনাটি — যা থেকে ঠিক বুট করার সময়েই কম্পিউটারের স্ক্রিনে এগুলো ফুটে ওঠে। এই রম-চিপগুলোয় অনেকক্ষেত্রে আবার প্রয়োজন পড়লে লিখে রাখা নির্দেশগুলো বদলে ফেলা যায়। তবে এই প্রক্রিয়াটা সবসময়ে খুব সহজসাধ্য নয়। যেমন বায়োস নির্দেশকে বদলানো — যার টেকনিকাল নাম বায়োস ফ্ল্যাশিং। যাকগে, এগুলো আমাদের পাঠমালার আলোচ্য বিষয় নয়।

০.৫.২।। উদ্বায়ী স্মৃতি — র‍্যাম

এই দ্বিতীয় ধরনের মুন্ডুতে কম্পিউটার তার স্মৃতিকে রাখে অস্থায়ী রকমে। পুনর্নবীকরণযোগ্য ডিসপোজেবল স্মৃতি ভরে রাখার মুন্ডু বলে ডাকা যায় র‍্যামকে। মেশিনে বিদ্যুত অফ হল তো এই স্মৃতিও হাফিজ হয়ে গেল। বা কম্পিউটার রিস্টার্ট করলেও। একধরনের ইলেকট্রনিক চিপ যাতে তথ্য লেখা এবং পড়া যায়। যেহেতু কম্পিউটার খুব সহজেই খুব দ্রুত এই চিপের যে কোনো জায়গায় লিখতে পারে বা যে কোনো তথ্য পড়ে ফেলতে পারে, বা

এখানে নতুন তথ্য লিখে নিতে পারে পরে পড়বে বলে, মানে এলোপাথাড়ি ভাবেই সক্রিয় করে তুলতে পারে এর যে কোনো অংশকে — তাই একে র‍্যানডম অ্যাকসেস মেমরি বলে।

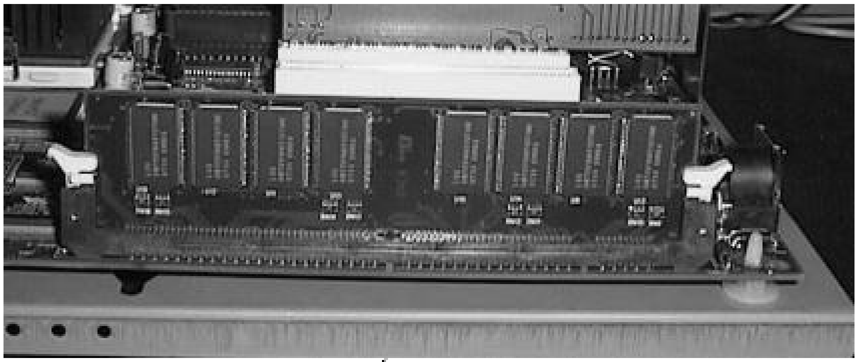
পিসির মাদারবোর্ডে র‍্যামচিপ লাগানোর জন্যে বিশেষ খাপ বা স্লট থাকে। সেই দুপাশে ক্লিপ লাগানো খাপের ভিতর র‍্যাম চিপের কার্ডটাকে লাগিয়ে দেওয়া হয়। একটা কার্ডে একগুচ্ছ করে র‍্যামচিপ থাকে। অনেকগুলো করে র‍্যামচিপ

লাগানো এই কার্ডগুলোকে ডাকা হয় মেমরি মডিউল বলে। মেমরি মডিউল অনেক রকমের হয়। তবে একটার জায়গায় অন্যটা লাগানো যায়না। মেমরি বাড়ানোর প্ল্যান থাকলে এইটা ভেবে কিনতে হয়। একই রকমের মডিউলের ভিতরেও আবার নানা ধরনের ব্যাপার আছে। তার অনেকগুলো বেশ ঝঞ্ঝাটের। সচরাচর চারদিকে যেরকম সব উইনডোবাজদের দেখি — তাদের মধ্যে একজন, আমার পরিচিত, ওর মেশিনের মাদারবোর্ড ৮১০, সবচেয়ে জনপ্রিয় এবং ঘাটিয়া কার্ডগুলোর একটা, একদিন জিগেশ করলাম, তোর মেশিন কেমন চলছে, বলল — ভালোই তো, আজ সকালেও তো খুলে গান শুনলাম, মাঝেমধ্যে অবশ্য একটু হ্যাং করে। আপনার যদি কম্পিউটারের কাছে চাহিদা নিছক এটুকুই হয় — গান চালানো মাঝে মধ্যে তো ক্যাসেটের ফিতে জড়িয়েও থেমে যায়, কম্পিউটার হ্যাং করলেই বা কী হয় — তাহলে আপনি এই পাঠমালার পাঠক না, আপনি এই পাঠশালার এই আমাদের মত পাঠাদের চেয়ে অনেক বেশি বুদ্ধিমান, জীবনটা কাটিয়ে দেবেন চুয়িংগাম চিবোতে চিবোতেই। কিন্তু কম্পিউটার ব্যাটাকে তার সম্ভাব্য ক্ষমতার কাছাকাছি চালাতে গেলেও হার্ডওয়ারটা একটু বুঝতে হবেই।

আমাদের কলকাতা লাগ-এ (ilug-cal@ilug-cal.org) অনেক সত্যিকারের কম্পিউটার জানা লোক আছে। সবার সঙ্গে আমার ব্যক্তিগত পরিচয়ও নেই, তবে তাতে কিছু এসে যায়না। আমার প্রেতমুম্বিত ডট-ম্যাট্রিক্স প্রিন্টারের (Epson LQ 300+) কোনো লিনাক্স ড্রাইভার নেই। লাগ-এ নালিশ করা মাত্রই, লাগ-এর মানস লাহা যেটা করেছিলেন সেটাকে বলা যায় নেটিউশনি — তাও ফি ছাড়া। দিন ছয়েকের গোটা পনেরো ইমেলের একটা টিউটোরিয়াল চলল। একটা ব্যাশ-স্ক্রিপ্ট লিখে কাঁচা ডেটার লেভেলে গোস্টস্ক্রিপ্ট বলে একটা সফটওয়ার দিয়ে প্রিন্টারটাকে চালানো হল — দিব্য চলছে — এই যে লিখছি, এই অসমাপ্ত মালটার তো একটা প্রিন্ট নিলাম, প্রেত কেন কোনো প্রেতিনীর চুম্বনেরও কোনো চিহ্ন নেই।

লাগ-এর অরিজিত হার্ডওয়ারটা বেশ ভালো বোঝে। আমি কোনো যন্ত্রপাতি কেনার বা সারানোর আগে ওর সাথে কথা বলে নিই। একটা বই রেফার করেছে — থম্পসন অ্যান্ড থম্পসন-এর পিসি হার্ডওয়ার ইন এ নাটশেল। চমৎকার একটা যন্ত্রপাতির ম্যানুয়াল। আপনাদের কারোর কোনো প্রয়োজন হলে লাগ-এ লিখবেন। এক লিনাক্সী অন্য লিনাক্সীর জন্যে দিবারাত্রি সমর্পিতমেল। এই লেখাটা পড়তে পড়তেও যদি কোনো কিছু বলার থাকে আমাদের জিএলটির ইমেল আইডিতে তো লিখতেই পারেন — সরাসরি আমাকেও লিখতে পারেন paagol@softhome.net। উত্তর পাওয়ার ব্যাপারে কোনো আশা রাখবেন না, আমি সরাসরি লিনাস টরভাল্ডসের নিজের হোমপেজ থেকে কোট করছি — কোটি বাতেলার সেরা বাতেলা সেটা — If you have any great suggestions, feel free to mail me,

and I'll probably feel free to ignore you । ছোট্ট একটা হোমপেজ — নিজের কোনো ছবি নেই — নিজের মেয়ের দুটো গেঁড়ি বয়সের ছবি, তার উপরে হেডিং 'লিনাস ভার্শন ২.০' — একবার ব্রাউজার-এ গুগলি মেরে দেখুন, লিনাস টরভাল্ডস দিয়ে সার্চ দিন।



র‍্যাম — মাদারবোর্ডে লাগানো অবস্থায়

যাকগে, যা বলছিলাম, জয় শ্রীর‍্যাম। এখানে একটা কথা খেয়াল রাখুন — এর পরে র‍্যাম তথা ভৌতিক স্মৃতি বা ভার্চুয়াল মেমরি নিয়ে অনেক বেশি ডিটেইলস নিয়ে আলোচনা আছে, পাঁচ এবং ছয় নম্বর দিনে — এই রম আর র‍্যাম — এদুটোর কাজ করার গতি কিন্তু তথ্য তুলে রাখার জায়গা মানে হার্ডডিস্ক-ড্রাইভ, সিডিরম-ড্রাইভ ইত্যাদির চেয়ে অনেক বেশি। সিডিরম, মানে একবার পুড়িয়ে ফেলা সিডি, সেটাও রম, কিন্তু বদলানো সহজ, একটা সিডি বার করো, আর একটা ঢোকাও। আর হার্ডডিস্কেও তাই, একটা ফাইল মুছে আর একটা লেখো, এখানে লেখা স্মৃতি বদলানোটা তাই কোনো ইশু না, বিষয় হল এদের গতি — কতটা দ্রুত কে কাজ করতে পারে। র‍্যাম-চিপ, রম-চিপ অনেক বেশি দ্রুত, দামও বেশি। কম্পিউটার জগতের প্রায় সমস্ত ঘটনারই মূল নিয়ন্তাগুলোর একটা এই দাম।

ক্রমে আরো দ্রুত সিপিইউর সঙ্গে তাল রাখতে গিয়ে মেশিনে ব্যবহৃত র‍্যামের পরিমাণ বাড়ছিল। আমার ৪৮৬ মেশিনে ছিল ১৬ এমবি, কাজের জন্যে যথেষ্ট, কখনো সমস্যা বলে মনে হয়নি। এখন এই অ্যাথলন ১৮০০ এক্সপি প্লাস মেশিনে ২৫৬ এমবি নিয়েও প্রায়ই মনে হয়, আর একটু হলে ভালো হত, আরো গ্নু-লিনাক্সে মেমরির ব্যবহারটা এত ভালো। সত্যিই এত হাতে গরম ফলাফল পাওয়া যায় র‍্যাম বাড়ানোর। উইনডোজ-এ, খুব বেশি র‍্যাম মেশিনকে বরং স্লো করে দিতে পারে, সিপিইউ যখন রাশি রাশি র‍্যাম-এর পাতার পর পাতা ধারণক্ষমতার মধ্যে গিয়ে পথ হারিয়ে ফেলে। এর কারণটায় আমরা পরে আসব — উইনডোজ আসলে তার মৌলিক গঠন মোতাবেক মাল্টি-টাস্কিং সিস্টেম নয়, মাল্টি-টাস্কিং বা বহু-কাজীপনাটা উপর থেকে গুঁজে দেওয়া। প্রক্ষিপ্ত। চার নম্বর দিনে এটা ধরব আমরা। মাল্টিটাস্কিং ব্যাপারটা কী, এবং সেটা কী ভাবে চালু হয়েছিল কম্পিউটার অপারেটিং সিস্টেমে।

র‍্যামের পাশাপাশি আর একজন সহযোদ্ধা ক্যাশে (cache) মেমরি। র‍্যামের কাজের মধ্যের মধ্যের ফাঁক গুলোয় তাপ্পি মারার কাজ করে ক্যাশে। পরে অনেক ভালো করে বুঝব কম্পিউটারের পাঁচ রকম স্মৃতিকে, হার্ডডিস্ক, র‍্যাম, এক্সটার্নাল ক্যাশে, ইন্টার্নাল ক্যাশে এবং সিপিইউ-র নিজের ভিতরের রেজিস্টার। সবচেয়ে শ্লথ হার্ডডিস্ক, সবচেয়ে দ্রুত সিপিইউ রেজিস্টার। এগুলো এই মুহূর্তে ল্যাটিন লাগতে পারে, পরে স্পষ্ট হয়ে যাবে। ক্যাশে-র প্রয়োজনীয়তাটা এসেছে সিপিইউ-র গতি বেড়ে যাওয়ার সঙ্গে তাল মিলিয়ে। এই ক্যাশে বা একটু দ্রুততর কুচো র‍্যাম জুড়ে দেওয়া হয় মেশিনে — স্মৃতির অভাবে যাতে বসে বসে দাঁত দিয়ে নখ কাটতে না হয় সিপিইউর। ৪৮৬-র পর থেকে ক্রমে আরো উন্নত সিপিইউগুলোতে এই ক্যাশে দেওয়া থাকে আসলে সিপিইউ চিপেই। ইন্টারনাল ক্যাশে। কিছুটা দেওয়া থাকে মাদারবোর্ডে — এক্সটারনাল ক্যাশে।

০.৫.৩।। তথ্য রাখার ভাঁড়ার

আমরা একটু আগে যে তিন ধরনের স্মৃতির কথা বলেছিলাম তার মধ্যে তিন নম্বর হল এই তথ্যের ভাঁড়ারঘর। মানে ডিস্ক। হার্ড ডিস্ক, বা সিডি মানে কম্প্যাক্ট ডিস্ক, বা ফ্লপি ডিস্ক। বা এমনকি ক্যাসেট ড্রাইভ বা জিপ ড্রাইভ-ও হতে পারে। ক্যাসেট ড্রাইভ মানে যেখানে একটা ক্যাসেট রেকর্ডারই ড্রাইভ হিশেবে ব্যবহার হয়, তথ্য তুলে রাখা হয়

ক্যাসেটের ফিতের চৌম্বক তলে, এবং পরে প্রয়োজনমত পড়া হয় ওখান থেকেই। আজকাল আর ব্যবহার হয় বলে মনে হয়না, প্রচুর প্রচুর বিরাট পরিমাণ তথ্য রাখার জন্যে বরং জিপ ড্রাইভ ব্যবহার করা হয়।

সিডি ড্রাইভ এবং সিডি ক্রমশ আরো বেশি বেশি করে ব্যবহার হচ্ছে তথ্য রাখার, ঢোকানোর এবং বার-করার মাধ্যম হিশেবে। তথ্য বার করার জন্যে ব্যবহার করতে গেলে ড্রাইভটাকে বার্নার ড্রাইভ হতে হয় — মানে সিডির তলে লেজার রশ্মি দিয়ে

পুড়িয়ে তথ্য লিখে রাখার মত সরঞ্জাম লাগে। একটা সিডিতে তথ্য রাখা যায় ৬৫০ মেগাবাইটেরও বেশি। তুলনা করুন একটা ফ্লপির সঙ্গে — ১.৪৪ মেগাবাইট। আর সিডিতে তথ্যের ত্রুটি ঘটা অনেক অনেক কঠিন। এবং একটা সিডির দাম এখন একটা ফ্লপির চেয়ে কম। শুধু এই কম দামের সিডিগুলোয় একটা সেট তথ্যই রাখা যায়, মানে লেখা তথ্য মুছে ফের লেখা যায়না। যা যায় এই ঢের। প্রথমদিককার সিডি একবারই পোড়ানো যেত, তারপর এলো রিরাইটেবল সিডি, যা বারবার পোড়ানো যায়, তারপর এখন তো এদের জায়গায় আসছে ডিভিডির রাজত্ব। সিডিতে তথ্য লেখা এবং পড়ার জন্যে বিশেষ রকম একটা ফাইল ব্যবস্থা লাগে, আইএসও ৯৬৬০, তার বিশদ আলোচনা করব আমরা আট নম্বর দিনে গিয়ে।

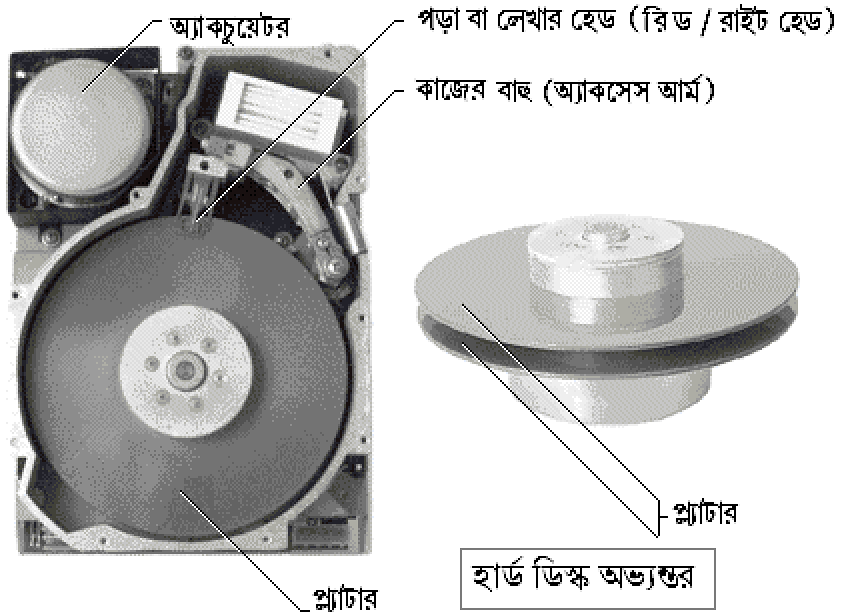
ভাঁড়ার মানেই এমন একটা জায়গা, যেখানে ইচ্ছেমতন তথ্য তুলে রাখা যায়। স্বেচ্ছাস্মৃতি। যতক্ষণ চাইব ততক্ষণই থাকবে এই স্মৃতি। যখন চাইব বদলে ফেলব। রাখতে চাইলাম তো রেখে দিলাম, চাইলাম না তো ফাইলগুলো ডিরেক্টরিগুলো উড়িয়ে দিলাম। অন্যগুলোয় এটা আগে থেকেই ছিল, রিরাইটেবল সিডি বেরোনোর পর থেকে সিডির বেলাতেও এই একই ভাবে সত্যি। এবং এদের প্রত্যেকের বেলাতেই এটা সত্যি যে মেশিনে কারেন্ট থাকুক আর না-থাকুক তাতে এই তুলে রাখা স্মৃতির কিছু এসে যায়না।

ইচ্ছেমতন, হাতের পাশে বইয়ের তাকে যেমন, তুলে রাখলাম, নামালাম, যখন যেমন, এই ভাঁড়ারগুলোর সুবিধেটা এইখানে। কিন্তু এদের খারাপটা হল এদের গতি। ওই রম বা র‍্যাম চিপের তুলনায় বহু বহু কম। আধুনিক দ্রুত সিপিইউর নিরবচ্ছিন্ন ভাবে কাজ করে চলার নিরিখে প্রায় অসম্ভব রকমের কম। তথ্য পড়া এবং লেখাতেই এতটা সময় চলে যায় যে সিপিইউ বেশিরভাগ সময়টা প্রায় বসেই থাকে। তার মানে, কম্পিউটারের কেন্দ্রীয় যে সত্তাটা, যাকে ঘিরে আর সব কিছু, সেই সিপিইউকেই আমরা তার পূর্ণ ক্ষমতার ধারেকাছেও ব্যবহার করে উঠতে পারিনা এই ডিস্ক ড্রাইভগুলোর দৌলতে। যে কারণে আজকাল ক্রমে বেশি আরপিএম (RPM — **R**otation-**P**er-**M**inute) আছে এমন হার্ড ডিস্ক ক্রমে জনপ্রিয় হচ্ছে আমাদের এখানেও। নইলে এমন একটা পরিস্থিতি তৈরি হয় যেখানে সেই কুকুরের ল্যাজ হাতে প্রেতযোনির মত সিপিইউ কাজের প্রতীক্ষায় ছটফট করছে কিন্তু শ্লথগতি ডিস্কড্রাইভ যথেষ্ট তাড়াতাড়ি তথ্য পড়ে বা লিখে উঠতে পারছেনা। আপনার মাথায় আগামী সাত লাইন তৈরি হয়ে গেছে, কিন্তু আপনার হাত অনেক পিছনে, লিখতে তার অনেক সময় লাগছে। এর টেকনিকাল নাম আইও-বটলনেক।

তথ্য রাখার এই জায়গাগুলোর তুলনায় র‍্যাম বা রম চিপ কাজ করে হাজার হাজার গুণ বেশি গতিতে। এই র‍্যাম আবার সিপিইউ-র তুলনায় অনেক অনেক শ্লথ। তাই সিপিইউকে কাজ করানোর জন্যে এই ড্রাইভগুলো থেকে তথ্য

তুলে র‍্যামে নিয়ে এসে জড় করে রেখে দেওয়া হয়। শুধু র‍্যামে না, পরে আমরা দেখব ভারচুয়াল মেমরি নামে হার্ডডিস্কেরই কিছু জায়গায়। গতিবেগের এই বিপুল তফাতের জন্যে এই তৃতীয় ধরনের মেমরিকে আমরা মেমরি বলে না ডেকে স্টোরেজ বা ভাঁড়ার বলে ডাকি। তথ্য রাখার একটা ভাঁড়ারঘর। মেমরি হল যেখান থেকে দ্রুত তথ্য পড়া বা লেখা যায় — র‍্যাম বা রম। আর ভাঁড়ার বা স্টোরেজ হল দীর্ঘ সময় ধরে তথ্য রেখে দেওয়ার জায়গা।

হার্ড ডিস্কের মধ্যে থাকে এক বা একাধিক শক্ত চাকতি (প্ল্যাটার) যাদের উপর একটা চৌম্বক আস্তরণ ফেলা থাকে। এই আস্তরণটাতেই তথ্য লিখে রাখা হয় বা পরে পড়া হয়, লেখা/পড়ার বিশেষ ছুঁচ বা রিড/রাইট হেড দিয়ে — হেডটা নড়ে এই



প্ল্যাটার-তলের খুব কাছ দিয়ে — ঘনিষ্ঠতাটা মোটামুটি এক ইঞ্চির আড়াইশো লক্ষ ভাগের এক ভাগ। এত নৈকট্যেও স্পর্শ না-করার সংযম দেখিয়ে হেডটা নড়ে বেড়ায় মিনিটে ৫৪০০ বার বা তারও বেশি গতিবেগে ঘূর্ণমাণ প্ল্যাটারপুঞ্জের তলে তলে, ডালে ডালে, পাতায় পাতায়। হেডের নড়াটাকে নিয়ন্ত্রণ করে কাজ করার বাহু বা অ্যাকসেস আর্ম। প্ল্যাটারটা বা প্ল্যাটারগুলো ঘোরে একটা স্পিন্ডল মোটরের সাহায্যে। এই স্পিন্ডল মোটর, হেড, অ্যাকসেস আর্ম এবং প্ল্যাটার সহ গোটাটা একটা চৌকো কৌটোয় ভরা থাকে — যাকে আমরা হার্ড ডিস্ক ড্রাইভ বলি। আমাদের এই পাঠমালার ছয় সাত এবং আট নম্বর দিনে বারবার ফিরে ফিরে আসবে হার্ডডিস্কের এই ভৌত জ্যামিতি। ফাইলসিস্টেম পার্টিশন ইত্যাদি বুঝতে আমাদের ভীষণভাবে কাজে লাগবে।

এই ভাঁড়ারটা হয় মেমরির চেয়ে আকারে বহুগুণ বড়। যেমন আমার মেশিনে দুটো হার্ড ডিস্কে তথ্য রাখার ভাঁড়ারে মোট জায়গা চল্লিশ গুণ দুই সমান আশি গিগাবাইট। আর র‍্যাম, আগেই তো বললাম, দুশো ছাপ্পান্ন মেগাবাইট। এই মেগাবাইট গিগাবাইট বাইট এগুলো আমরা একটু বাদেই ভালো করে বুঝব। আপাতত জেনে রাখা যাক এক গিগাবাইট হল ১০২৪ মেগাবাইট। এক মেগাবাইট হল ১০২৪ কিলোবাইট। এক কিলোবাইট মানে ১০২৪ বাইট।

০.৬।। তথ্য নিয়ন্ত্রণ বা কন্ট্রোল ডিভাইস

আগেই বলেছি কম্পিউটারের মূল কাজ হল ডেটা প্রসেস করা, আমরা যাকে তথ্য চটকানো বলে ডেকেছি। কিন্তু কম্পিউটার তো সেটা আপনা থেকে এমনি এমনি করেনা, তাকে সেটা করাতে হয়। তার জন্যে তাকে নির্দেশ দিতে হয়। নির্দেশ যায় কম্পিউটারের বিভিন্ন অংশগুলোর কাছে যাতে তারা তাদের প্রত্যেকের ভূমিকা পালন করে — নইলে মোট কাজটা হতে পারবে না। ইনপুট ডিভাইস ব্যবহার করা মানে তথ্য ঢোকানোর যন্তরগুলোকে বলতে হবে, ঢোকাও বাবাসকল, যাতে তারা চটকানোর তথ্যগুলোকে প্রথমেই ঢুকিয়ে আনে। শুধু চটকানোর তথ্য না, কী ভাবে চটকাবে সেই রেসিপিটাও। এবার যে তথ্য চটকাতে হবে আর যে নির্দেশ অনুযায়ী চটকাতে হবে — এই দুই ধরনের তথ্যকে কোথাও চট করে হাতের কাছে পাওয়ার মত জায়গায় তুলে দেওয়ার নির্দেশও দিতে হবে সিস্টেমকে, নইলে

সে কাজ করতে গিয়ে বারবার তো দিশাহারা হয়ে পড়বে। সচরাচর এই চট করে তুলে রাখা এবং চট করে ফিরে পাওয়ার জায়গাটা হল পিসির মূল মেমরি। এর মধ্যে আবার বহু ছোট ছোট নির্দেশের টেকনিকালতা আছে। আছে তার ভিতর আগে পরে-র সময়রেখা বা ক্রোনলজি। ঠিক ঠিক প্রথা অনুযায়ী, হার্ডওয়ারের নিজস্ব ব্যাকরণ অনুযায়ী এই নির্দেশগুলোকে আবার নিয়ন্ত্রণ করতে হবে।

নির্দেশ এবং নিয়ন্ত্রণের এই গোটা কাজটা কম্পিউটারের যে অংশটা করে তাকে বলে কন্ট্রোল পোর্শন বা নিয়ন্ত্রক অংশ। এই কন্ট্রোল পোর্শনের প্রায় গোটাটাই থাকে সিপিইউ চিপে। সিপিইউ চিপের মধ্যে ভরা থাকে চটকানোর আর নিয়ন্ত্রণের, প্রসেসিং-এর আর কন্ট্রোলের দু-ধরনের যন্ত্রপাতিই। এই দু ধরনের বস্তু ছাড়া আরো একটা বস্তু থাকে এই সিপিইউ চিপে — কম্পিউটারের দ্রুততম মেমরি — সিপিইউ রেজিস্টার, এবং দ্বিতীয় দ্রুততম মেমরি — ক্যাশে। আরো ভাল করে এটা আমরা পরে বুঝব। কন্ট্রোল পোর্শন বা নিয়ন্ত্রক অংশের একটুখানি থাকে সিপিইউ চিপের বাইরে — সেটা হল চিপ-সেট আর এমবেডেড কন্ট্রোলার চিপ। এদুটো নিয়ে আমরা আলোচনা আমাদের এই গোটা পাঠমালাতেই আনবনা। এগুলো নিয়ে আলোচনা করার যোগ্যতাও আমার নেই । তবে এই নিয়ন্ত্রণ অংশ নিয়ে আলোচনায় আমাদের বারবার ফিরে ফিরে আসতে হবে। তার হার্ডওয়ার তার সফটওয়ার, তার লুপ, তার প্রোগ্রামিং ইত্যাদি। তিন নম্বর দিনে আডা লাভলেসের আলোচনায় আমরা কম্পিউটার চিন্তায় এর প্রাচীনতম চেহারাটাকে দেখতে পাব। আর এই বইয়ের মাপে সবচেয়ে জটিল রকমে এই আলোচনাটা আসবে আমাদের শেষতম মানে দশ নম্বর দিনে, প্রায় শেষ অংশে, সেখানে এই নিয়ন্ত্রণের সঙ্গে ব্যাশ শেলের সম্পর্কটাকে দেখব আমরা।

এই কন্ট্রোল পোর্শনটাকে ছেঁটে দিলে একটা পিসি কিন্তু একটা কোট-টাই-পরা ক্যালকুলেটরের চেয়ে বেশি কিছু নয়। ক্যালকুলেটরেও তো বেড়ে তথ্য ঢোকানো যায় — ০ থেকে ৯ অব্দি অঙ্কের চাবি আর +, −, ×, ÷ — এই অপারেশন চাবিগুলো দিয়ে। এবং তথ্য বার করেও চমৎকার, এবং বেশ তাড়াতাড়ি। যে হিশেবটা করছে সেটা করতে যত গাবদা গাবদা সংখ্যা যতক্ষণ ধরেই লাগুক — সেই পুরো সময়টা জুড়ে ততটা ধরে রাখবার মত মেমরিটা এর আছে। এবং ততটা প্রসেসিং ক্ষমতা এর নিশ্চয়ই আছে যতটা থাকলে ওই হিশেবগুলো করা যায়। তাহলে ক্যালকুলেটরটার কী নেই যা একটা পিসির আছে? তফাতটা এই যে প্রতিটি বোতামই নিজে টিপে দিতে হবে প্রত্যেকবার — নিজের কাজের বোতাম নিজে নিজে টিপে নেওয়ার ক্ষমতাটা ক্যালকুলেটরটার নেই। যেকোনো প্রোগ্রামেবল ক্যালকুলেটরকে একটা দূর অব্দি, কয়েকটা স্টেপ-এর হিশেব অব্দি, নিজের বোতাম নিজে টিপে নেওয়ার ব্যবস্থা করানো যায় — এবং সেই দূরত্বটা অব্দি ক্যালকুলেটরটাকে একটা কুচো মাত্রার কম্পিউটার বলা যায়। কিন্তু ওই হাতে গোনা স্টেপকটাতে একটা চিকণতম সি প্রোগ্রামও তাতে দৌড় করানো যাবেনা। আসলে মাত্রাই তো সব। মাত্রার তফাতই লুচি আর বিষের একমাত্র তফাত। লুচি আর পটাশিয়াম সায়ানাইড তো একেবারে একই বিষ — দুটো খেলেই মানুষ মরে একইরকম ঠান্ডা হয়ে যায়। বিশ্বাস না হলে আপনি এক সিটিং-এ তেরোশো বাহাত্তরটা লুচি খেয়ে দেখুন — ওর চেয়ে কম খেলেও আপনি নিশ্চিতভাবেই মরবেন কিনা আমি বলতে পারবনা, সরি, কখনো খেয়ে দেখিনি। খেয়ে দেখুন — মরে গেলে জানাবেন। তেরোশো বাহাত্তর ফিগারটা কমিয়ে দেব।

০.৭।। একটা সারণী — কাজ ভার্সাস উপাদান

সারণী শব্দটা এমন শক্ত কিছু নয় — ওর বাংলা হল টেবিল। একটা তালিকা তৈরি করা যাক এবার — এই শূন্য নম্বর দিনে একটা পিসির যে বিভিন্ন অংশগুলো আমরা আলোচনা করলাম — তার কোনটা কী বা কী কী কাজ করে।

এখানে দুটো জিনিষ মাথায় রাখার। এক, আমরা আমাদের এই আলোচনায় হাতে গোনা কয়েকটা মাত্র ধরনের যন্ত্রপাতিকেই এনেছি। যেগুলোয় আমরা সবচেয়ে বেশি অভ্যস্ত। বা, বলা ভাল, আমি। আমার মেশিন, আমার বন্ধুদের মেশিনে যেসব যন্ত্রপাতি দেখেছি। এই নিজের পছন্দমত জিনিষপত্তর নিয়ে লেখার অধিকারটা আমার আগাগোড়াই ছিল। কারণ, আমি কোনো টেক্সটবই লিখছি না। আমাদের মধ্যমগ্রাম জিএলটির সূত্রে যেসব ছেলেমেয়ের সঙ্গে কথা হয়েছে, যাদের পড়াতে শুরু করলাম এই লিনাক্স-হাতেখড়ি কোর্সে, তাও তো এটাই প্রথমবার, তাদের কাছে পৌঁছনোটাকে সহজ করার জন্যে এই লেখাটা শুরু। তাও তো এই ০ নম্বর লেখাটা মূল প্ল্যানে ছিলই না। শুরুই হচ্ছিল ১ থেকে। ৯ অব্দি হয়ে গেছিল। এর মধ্যে যারা এই সিরিজটা পড়ল তার ভিতর

সোমনাথ আর বুড়িয়া প্রবল চেঁচাল, কিছুই বুঝতে পারছেনা বলে। সেটা অবভিয়াসলি ওদের বুদ্ধির দোষ, আমার লেখার নয়। কিন্তু কী করি, লিখতেই হল এই ০ নম্বরটা।

আর দু নম্বর বিষয়টা কম্পিউটার শাস্ত্রের নিরিখে আর একটু জরুরি। এতক্ষণ অব্দি যে আলোচনাটা আমরা করে এলাম তাতে পিসি ছাড়া কোনো পিসেমশাইয়েরই নাম করিনি। কিন্তু এই পিসেমশাইরাই সংখ্যাগরিষ্ঠ কম্পিউটারের ইতিহাসে। তিন চার এবং পাঁচ নম্বর দিনে আমরা সেটা পড়ব। অন্যান্য প্রবীন কম্পিউটার বা পিসেমশাইদের তুলনায় পার্সোনাল কম্পিউটার বা পিসি অনেক অর্বাচীন। কিন্তু এখানেও মূলত ওই বুড়িয়া-সোমনাথদের মত পাঠকদের কথা ভেবেই পিসি ছাড়া আর কিছু আনলাম না। এবং একদম ছুঁলাম না অপারেটিং সিস্টেমকে। ইউনিক্স, লিনাক্স, উইনডোজ — কিছুই না। সেসব শুরু হবে ১ নম্বর দিন থেকে। কিন্তু এই তালিকাটা পড়ার সময়ে আমরা এটা মাথায় রাখব যে এটা বানানো পিসির উপাদানগুলোর নিরিখে। মেইনফ্রেম বা অন্য কম্পিউটারের বেলায় মাত্রার বদলটা এত নাটকীয় হয় যে এমনকি উপাদানগুলোর প্রাথমিক ভূমিকাটাও অনেকসময়ই একদম বদলে যায়। মাত্রা জিনিষটা কতটা জরুরি তাতো তিন প্যারা আগেই বললাম। এবার তালিকাটা দেখুন — এখানে বাঁদিকের স্তম্ভে পরপর আছে উপাদানগুলোর নাম। আর একদম উপরের সারিটায় কাজগুলোর নাম।

অংশের নাম	কাজ				
	তথ্য ঢোকানো	তথ্য বার-করা	তথ্য চটকানো	তথ্য ধরে-রাখা	তথ্য নিয়ন্ত্রণ
কিবোর্ড	✓	✕	✕	✕	✕
মাউস	✓	✕	✕	✕	✕
ট্র্যাকবল	✓	✕	✕	✕	✕
মাইক্রোফোন	✓	✕	✕	✕	✕
ফ্লপি ডিস্ক	✓	✓	✕	✓	✕
সিডি-আর	✓	✓	✕	✕	✕
সিডি-আরডব্লু	✓	✓	✕	✓	✕
হার্ড ডিস্ক	✓	✓	✕	✓	✕
জিপ ড্রাইভ	✓	✓	✕	✓	✕
সাউন্ডবক্স	✕	✓	✕	✕	✕
মনিটর	✕	✓	✕	✕	✕
প্রিন্টার	✕	✓	✕	✕	✕
আলু (সিপিইউ)	✕	✕	✓	✕	✕
ক্যাশে (সিপিইউ)	✕	✕	✕	✓❈	✕
কন্ট্রোল সার্কিট (সিপিইউ)	✕	✕	✕	✕	✓
মাদারবোর্ড চিপ সেট	✕	✕	✕	✕	✓
মাদারবোর্ড রম	✕	✕	✕	✓❈	✕
মাদারবোর্ড র‍্যাম	✕	✕	✕	✓❈	✕

✕ মানে এই উপাদানটা এই কাজ করেনা, ✓ মানে এই উপাদান এই কাজ করে, আর ❈ চিহ্নটা বোঝাচ্ছে যে এই ধরে-রাখাটা একটু আলাদা, হয়, নিতান্তই ক্ষণিকের অতিথি — একবার মেশিন বন্ধ হলেই উবে যাবে, নয়তো, ইচ্ছে হলেই বদলানো যাবেনা। এরা স্টোরেজ ডিভাইস বা ভাঁড়ার থেকে আলাদা, একটু আগে যাদের আমরা মেমরি বলে ডেকেছিলাম। স্টোরেজ বা ভাঁড়ার হল তারা, যেখানে রাখতে বা বাদ দিতে পারি আমরা যখন খুশি যেভাবে খুশি যত সময় খুশি।

০.৮।। কম্পিউটার এবং তথ্য

আগেই বলেছি বারদুয়েক, এবং এই পাঠমালা জুড়ে আরো বারংবার বলব, কম্পিউটার বেচারার ভাষাবোধ দরিদ্র বললেও কম বলা হয়। মালটা একদমই আকাঠ, ০ আর ১ ছাড়া কিছু বোঝেনা। যে কোনো তথ্যকেই কম্পিউটার বোঝে এই ০ আর ১ দিয়ে। এইভাবে কম্পিউটারকে কোনো তথ্য বোঝানোর একটা খুব গুরুত্বপূর্ণ দিক আছে।

এই ভাবে ০ আর ১ দিয়ে কোনো তথ্যকে বোঝানো মানে একভাবে সেটাকে সম্পূর্ণ রকমের নির্মল বা নয়েজ-মুক্ত বা ত্রুটিহীন করে তোলা গেল। কোনো তথ্যকে অনেক ভাবে রাখা যেতে পারে। যেমন ধরুন একটা গান। সেই গানের আওয়াজটাকে আমরা ম্যাগনেটিক একটা ফিতেয় তুলে রাখতে পারি। যেখানে কোনো নির্দিষ্ট পরিমাণ এলাকায় মোট চৌম্বক অনুর উপস্থিতির মাত্রা দিয়ে শব্দের বিভিন্ন প্রকারকে, তাদের বিভিন্ন ওঠানামাকে ধরে রাখা হয়। এই ভাবে শব্দ ধরে রাখতে গিয়ে আমরা গানের পাশাপাশি যে কী বিপুল পরিমাণ কোলাহলও ধরে রাখি সেটা আমরা সবসময় খেয়াল করিনা। খুব সহজ একটা এক্সপেরিমেন্ট করা যায়। একটা গানকে একটা ক্যাসেট থেকে অন্য ক্যাসেটে রেকর্ড করুন। দ্বিতীয়টার থেকে আবার প্রথমটায়। এরকম করে একশোবার করার পর শেষ ক্যাসেটটায় গানটাকে আর খুঁজে পাওয়া যাবেনা। মোট কোলাহল বা নয়েজ-এর পরিমাণ তখন গানের মূল আওয়াজকে ছাপিয়ে গেছে। একটা ক্যাসেটে অনেকদিন গান রেকর্ড হয়ে থাকার পর, প্রকৃতির প্রকোপে, আরো আমাদের ভিজে আবহাওয়ায়, বা, অনেকবার বাজানোর পর, এই ম্যাগনেটিক আস্তরণে আওয়াজ ধরে রাখার ব্যথাটা যেমন আমরা হাড়ে হাড়ে টের পাই।

কিন্তু কম্পিউটার একবার যখন সেই আওয়াজটাকে ০ আর ১-এ পর্যবসিত করে নিল, সেটার একটা মস্ত ভালো জায়গা চলে এল। এই ০ আর ১-এর পদ্ধতি, যার টেকনিকাল নাম বাইনারি, সেটা মেনেই কম্পিউটার এবার ফের শব্দটাকে বানিয়ে নেবে। সে জানে কোন বাইনারি কম্বিনেশন মানে কী আওয়াজ বানাতে হবে — সেটাই তার অডিয়ো বা মাল্টিমিডিয়া সফটওয়ারের কাজ। ধরুন ম্যাগনেটিক টেপেও যেমন ব্যবহারের বা কালের বলীরেখা পড়ছিল সেরকম পড়ল সিডির রেকর্ডেবল তলেও। এবার, যখন সে ০ আর ১ হিশেবে সিডির রেকর্ডযোগ্য আস্তরণে লেজারের পোড়ানো গর্তগুলোকে পড়তে যাবে তখন ০ এবং তার সমস্ত কাছাকাছি প্রতিক্রিয়াকেই সে ফের ০ করে দেবে। ০ থেকে তার ঠিক বিপরীত ১ হয়ে ওঠাটা প্রায় অসম্ভব। আবার ১-এর বেলাতেও তাই। অর্থাৎ, যথেষ্ট পরিমাণ মেকানিকাল বা যান্ত্রিক ত্রুটি সিডিটায় থাকলেও শেষ অব্দি তার ফলাফলটা ত্রুটিমুক্তই হবে।

এবার প্রশ্ন আসে তথ্যের মোট পরিমাণটাকে আমরা মাপব কী করে? একটা ব্যাপার আছে, খুব কম্পিউটার আসার আগে অব্দি খুব নিখুঁত ভাবে মাপার প্রয়োজনও পড়ত না, মাপাটা সম্ভবও ছিলনা। যেমন মনে আছে, এমএসসির পরপরই বারোমাস পত্রিকায় 'সুমিতার ডাকনামে মিনু' বলে একটা বড় গদ্য ছাপার আগে সম্পাদক অশোক সেন, উনি ইকনমিক্সে মাস্টারমশাইও ছিলেন, আমার কাছে জানতে চেয়েছিলেন, কত ওয়ার্ডের লেখা? এই প্রশ্নটাই যে হয় সেটা সেই প্রথম জানলাম। বেরিয়ে এসে আফসার আমাকে শেখাল, আরে দুতিন জায়গার দুতিনটে লাইন থেকে শব্দের সংখ্যাটা গুনে নাও, তারপর দুতিন জায়গার দুতিনটে পাতা থেকে লাইনের সংখ্যাটা গোনো, এবার শব্দ গুণ লাইন গুণ পাতা মানেই মোট শব্দ। আমার এইচএস থেকে স্ট্যাটিস্টিক্স ছিল, রাশিশাস্ত্রের এতাবৎ বীভৎসায় আমি শিউরে উঠেছিলাম, কিন্তু পরে দেখেছি, বেশ কাজে দেয়।

এখন তো কম্পিউটারের কল্যাণে এটা কোনো ব্যাপারই না, আরো আমার তো বাঁয় হাত কা খেল, কারণ আমি বাঁ হাতে মাউস চালাই, প্রথম থেকেই, ডান হাতটার জন্যে জরুরিতর কাজ থাকে, চায়ের কাপ সিগারেট ইত্যাদি। শুধু মাউস টিপেই এক নিমেষেই বলে দেওয়া যায়। যেমন ঠিক এই আগের দাঁড়িটা অব্দি এই লেখাটায় পাতা ২৩ (এ-ফোর সাইজ), শব্দ ৯৫৬৬, স্পেস-সহ অক্ষর ৬৯১৩৪, প্যারা ২০৫, লাইন ৭৬৭। এটা এখন মাপার দরকারও পড়ে অনেক। যেমন এটা লিখছি আর মনে মনে হায় হায় করছি এই ভেবে যে এই সিরিজের অন্য লেখাগুলোর মত (এই সিরিজের ১ থেকে ১০ ইতিমধ্যেই লেখা হয়ে গেছে, বলেছি) এটা লাগ-এর তথাগত সায়মিন্দু সঙ্কর্ষণ ইন্দ্র অরিজিতকে বা সৈকত জয়ন্তী অশেষ অশোকদাকে ইমেল করে পাঠাতে গেলে কান্না পাবে, কারণ, এইমাত্র একবার পিডিএফ করে দেখলাম, সাইজ এর মধ্যেই ১ এমবি অতিক্রম করে গেছে — অতোগুলো ছবি যে।

কতটা পরিমাণ সম্ভাব্য ভাঁড়ারে কতটা তথ্য আমায় রাখতে হবে এটা এখন একটা জরুরি প্রশ্ন। রাখা এবং একটা মেশিন থেকে আর একটা মেশিনে তাকে পাঠানো। বা, এমনকি, একটা প্রোগ্রাম একটা ফাইল খোলার সময় কতটা মেমরি তার জন্যে ছেড়ে দিতে হবে — ইত্যাদি। তাই মাপতে হবেই। কিন্তু মাপব কী করে?

০.৮.১।। তথ্যের পরিমাপ

একটা প্রশ্নের সবচেয়ে সরল উত্তর কী হতে পারে? হয় হ্যাঁ, অথবা না। ১, অথবা ০। এবং এই হ্যাঁ বা না দিয়ে উত্তর দেওয়া যায় এমন প্রশ্নের সঠিক উত্তর রাখতে ধরে রাখতে গেলে আমার কতটা মেমরি লাগবে? ঠিক ততটুকু জায়গা যেখানে রাখা যায় হয় একটা ১ অথবা একটা ০। ঠিক এইটুকু জায়গাকেই, যেখানে হয় একটা ১ রাখা যায় নয় একটা ০, ঠিক এটাকেই বলে বিট (BIT — **B**inary-dig**IT**) — কম্পিউটারের তথ্য পরিমাপের ক্ষুদ্রতম একক।

গণিতের হিশেবে একটা বিট-এর মান হতে পারে হয় ১ অথবা ০। এর মানে হতে পারে সত্যি/মিথ্যে, বা, হ্যাঁ/না, বা, আছে/নেই — এর যে-কোনোটাই। বৈদ্যুতিক এঞ্জিনিয়ারিং-এর নিরিখে এটাকে যান্ত্রিক কাঠামোয় এই রকম একটা আকারে পর্যবসিত করা যায় যে একটা সার্কিটে বিদ্যুৎ আছে না নেই। বা, সার্কিটে ভোল্টেজের মান উঁচু না নিচু? আবার যখন ক্যাসেট ড্রাইভে কম্পিউটার তার তথ্য তুলে রাখছে বা সেখান থেকে পড়ছে তখন এটা হতে পারে একটা বিশেষ চৌম্বক ক্ষেত্রের মেরুটা এইদিকে আছে না এর বিপরীত দিকে। এরকম অনেক ভাবেই এই শূন্য/এক-এর এক বিট তথ্যকে রাখা যেতে পারে।

এইভাবে তথ্য রাখা মাত্রই সবচেয়ে বড় যে জিনিষটা ঘটল তা হল, এখন থেকে সম্পূর্ণ গাণিতিক রকমের নিখুঁত ভাবে বলে দেওয়া যাবে তথ্যের পরিমাণ ঠিক কতটা। একটা প্রশ্নমালার উত্তর দিতে ঠিক কতগুলো বিট লাগবে। মানে ব্যাপারটা অনেকটা এরকম — একটা জটিল প্রশ্নকে অনেকগুলো সরল প্রশ্নে ভেঙে নেওয়া যাদের প্রত্যেকটাকেই একটা হ্যাঁ বা না দিয়ে উত্তর দেওয়া যাবে।

তথ্য পাঠানো বিষয়ে একটা পুরোনো গল্প এইরকম। একটা শহরে কোনো আক্রমণ হচ্ছে কিনা অনেক দূরে একটা উঁচু জায়গায়, একটা লাইটহাউসে বসে একজন পাহারা দিচ্ছে। সে যদি দেখে, আক্রমণকারীরা নৌকো করে সমুদ্রপথে আসছে তাহলে সে একটা আলো জ্বেলে দেবে। যদি দেখে তারা ঘোড়ায় চড়ে স্থলপথে আসছে তাহলে সে দুটো আলো জ্বেলে দেবে। তার মানে এখানে সম্ভাব্য উত্তর তিনটে।

(১) কোনো আলো জ্বলেনি — আক্রমণ এখনো হচ্ছে না — সংখ্যা ০

(২) একটা আলো — জলপথে আক্রমণ হচ্ছে — সংখ্যা ১

(৩) দুটো আলো — স্থলপথে আক্রমণ হচ্ছে — সংখ্যা ২

এই তিনটে উত্তর সংখ্যা দিয়ে রাখতে গেলে লাগবে ০/১/২। এই মাত্র তিনটে উত্তরেই হয়ে গেল কারণ এখানে তথ্যটা খুব একটা জটিল নয়। তার সম্ভাব্য সমাধানের সংখ্যা মাত্র তিন। ধরুন সম্ভাব্য উত্তরের সংখ্যা যদি ১০০ হয়ে যায় তখন এইভাবে আলো দিয়ে তথ্য পাঠানোটা অসম্ভব রকমের জটিল হয়ে পড়বে। এই সদ্য কালীপুজোর রাতে ছাদে প্রদীপ জ্বালানোর জ্যান্ত এবং টাটকা অভিজ্ঞতা থেকে বলছি, কয়েকটা আলো হলে একদিক থেকে জ্বালাতে জ্বালাতে অন্যদিক থেকে নিভে যেতে শুরু করবে। শেষে এনলাইটেনমেন্টহীন হতাশায় আমি স্থির করেছিলাম, এর চেয়ে অন্যদের ছাদে জ্বালানো আলো দেখাটাই সবচেয়ে ভালো। আরো লাইটহাউস, সমুদ্রের হাওয়া হু হু করে বইছে, ওতো জ্বালানোই যাবেনা।

০.৮.২।। আঙ্কিক তথ্যের আকার ও পরিমাণ

আমার ঠাকুর্দা আজ থেকে বছর ছয়েক আগে একশো তিন বছর বয়সে মারা গেছিলেন। বেঁচে থাকলে এখন স্কোর হত একশো নয়। ধরুন তিনি বেঁচে আছেন এবং এই ক্ষীণজীবী ক্ষণজীবী বিশ্বে তাঁর এই বিস্ময়কর বয়সটা আপনি নিজেই যাচাই করে নিতে চাইছেন। আপনি ধরে নিলেন, ঠাকুর্দার বয়সটা ১২৮-এর কম, তার অর্ধেক করলেন। ৬৪। এবার প্রশ্ন করলেন, আপনার বয়স কি ৬৪ বা তার বেশি?

উত্তর হল হ্যাঁ, — মানে ১।

এবার দেখলেন উপরের মানে ১২৮ থেকে ৬৪-র অবশিষ্টের অর্ধেক মানে ৬৪-র অর্ধেক হল ৩২। ৬৪ আর ৩২ হল ৯৬। এবার জিগেশ করলেন, আপনার বয়স কি ৯৬ বা তার বেশি?

উত্তর হল, হ্যাঁ — মানে ১।

এবার ৯৬-এর সঙ্গে যোগ করলেন উপরের অবশিষ্টের অর্ধেক মানে ১৬। ৯৬ আর ১৬ হল ১১২। জিগেশ করলেন, আপনার বয়স কি ১১২ বা তার বেশি?

উত্তর হল, না — মানে ০।

এবার ৯৬ এবং ১১২-র যে দূরত্ব মানে ১৬, তার অর্ধেক মানে ৮ যোগ করলেন ৯৬-এর সঙ্গে। পেলেন ১০৪। জিগেশ করলেন, আপনার বয়স কি ১০৪ বা তার বেশি?

উত্তর হল, হ্যাঁ — মানে ১।

এবার ১০৪ আর উপরের মান ১১২-এর মধ্যের ফারাক মানে ৮-এর অর্ধেক মানে ৪ যোগ করলেন ১০৪-এর সঙ্গে। পেলেন ১০৮। জিগেশ করলেন, আপনার বয়স কি ১০৮ বা তার বেশি?

উত্তর হল, হ্যাঁ — মানে ১।

এবার ১০৮ আর ১১২-র ফারাক ৪-এর অর্ধেক মানে ২ যোগ করলেন ১০৮-এর সঙ্গে। জিগেশ করলেন, আপনার বয়স কি ১১০ বা তার বেশি?

উত্তর পেলেন, না — মানে ০।

এবার ১০৮ আর ১১০-এর ফারাক মানে ২-এর অর্ধেক মানে ১ যোগ করলেন। জিগেশ করলেন, আপনার বয়স কি ১০৯ বা তার বেশি?

উত্তর পেলেন, হ্যাঁ — মানে ১।

এবার পরপর এই অঙ্কগুলো লিখে ফেলুন — ১১০১১০১। এই উত্তরটা মনে রাখুন। এটা একটা বাইনারি সংখ্যা। এখানে এককুচো একটু বাইনারি আর ডেসিমাল নিয়মের তফাতটা বুঝে আসা যাক।

০.৮.২.১।। এইরে, অঙ্ক কষাচ্ছে

বাইনারি মানে যেখানে অঙ্ক বা ডিজিটগুলো, যেগুলোকে মিলিয়ে আমরা সংখ্যা বা নাম্বার পাই — তারা হল ০ আর ১। বাইনারি হল এই দুই অঙ্কের সিস্টেম। যেমন আমরা দশমিক বা ডেসিমাল প্রক্রিয়ার সঙ্গে অভ্যস্ত। যেখানে ডিজিটের সংখ্যা দশ। ০, ১, ২, ৩, ৪, ৫, ৬, ৭, ৮, এবং ৯।

এবার এই দশমিক পদ্ধতিতে লেখা একটা সংখ্যাকে ভাবুন — ধরুন ৩৪৫। মুখে কী ভাবে পড়লেন? তিনশো পঁয়তাল্লিশ। তিনশো, চল্লিশ, পাঁচ।

$$
\begin{aligned}
৩৪৫ &= (৩০০ + ৪০ + ৫) \\
&= ((৩ \times ১০০) + (৪ \times ১০) + (৫ \times ১)) \\
&= ((৩ \times ১০^{২}) + (৪ \times ১০^{১}) + (৫ \times ১০^{০}))
\end{aligned}
$$

তার মানে ৩৪৫ এই সংখ্যাটার প্রতিটি অঙ্ক গুণ হয়ে যাচ্ছে ১০-এর কোনো একটা পাওয়ার দিয়ে — ০, ১, ২, ইত্যাদি। যত বড় সংখ্যা হবে এই গুণক-টার সূচক বা পাওয়ারটা এক এক করে বেড়ে যাবে। এবং এই দশ সংখ্যাটা কেন হল এখানে? কেন এটা অন্য কোনো সংখ্যা নয়? কারণ, আমাদের এই কাঠামোয়, ডেসিমাল সিস্টেমে বা দশমিক পদ্ধতিতে অঙ্কের সংখ্যা দশ। যুক্তিটা একটু খেয়াল করুন। প্রথমে একদম ডানদিকের ঘর — মানে এককের ঘর — যেখানে ১ দিয়ে গুণ করছি। সেখানে ০ থেকে ৯ অব্দি দশটা সংখ্যা পরপর আসছে, সংখ্যাটা এক এক করে বাড়ছে। যেই ৯-এ পৌঁছে গেলাম, আর বাড়ানোর উপায় নেই। ঘর তো মাত্র একটা। সঙ্গে সঙ্গে পরের মানে দশকের ঘরে ১ বসালাম, আর এককের ঘরে এখন কিছু নেই, তাই ০। মানে সংখ্যাটা হল ১০। আবার বাড়াচ্ছি সংখ্যাটাকে এক এক করে। বাড়াতে বাড়াতে ১১ থেকে ১৯। এবার কিন্তু বাড়ানোর উপায় আছে আরো অনেকক্ষণ। দশকের

ঘরটাতে ১-কে ২ করে দিলাম। আবার এককের ঘরে ০। এই করে চলল ৯৯ অব্দি। যখন এককের ঘরে ৯, দশকের ঘরে ৯। আবার স্পেস শর্টেজ। মানে পরের ঘরে যাও কাকা। পেলাম ১০০। যার শতকের ঘরে দিলাম ১। দশক আর এককে পড়ল ০। এইভাবে চলতেই থাকবে।

এবার ভাবুন এককের ঘরে একটা অঙ্ক বাড়ানো মানে আমি বাড়াচ্ছি এক এক করে। তাই গুণক হল ১ মানে $১০^{০}$, কিন্তু দশকের ঘরে একটা অঙ্ক এক বাড়ানো মানে তার পিছনে বাড়তে পারছে এককের ঘরে ০ থেকে ৯ মানে মোট দশটা সংখ্যা। তার মানে এই ঘরে ১ বাড়ানো আসলে ১০ বাড়িয়ে দেওয়া। তাই দশকের ঘরে মানে দ্বিতীয় ঘরে গুণক ১০ মানে $১০^{১}$। ঠিক এই একই যুক্তিতে শতকের মানে তৃতীয় ঘরে ১ বাড়ানো মানে আসলে তার পিছনে দশকের ঘরে ০ থেকে ৯ এই দশটা সংখ্যা বাড়ানো। মানে, ঠিক আগের যুক্তিতে, এককের ঘরে ১০ গুণ ১০ মানে ১০০টা সংখ্যা বাড়ানো। কারণ, দশকের ঘরে ১ বাড়া হল এককের ঘরে ১০ বাড়া। ০ থেকে ৯৯ এই একশোটা সংখ্যা বাড়তে দেওয়া। তাই এই ঘরে গুণক ১০০ মানে $১০^{২}$। এই ভাবে প্রতিটি ঘরে গুণকটা বাড়বে ১০ এর পাওয়ার বা সূচকে এক এক করে।

এবার ঠিক এই যুক্তিটাকে ডিটো কপি করে দিন বাইনারি সিস্টেম বা দ্বিত্ব পদ্ধতিতে। শুধু ডেসিমালের জায়গায় বাইনারি। মানে ১০ এর জায়গায় ২। মানে ০, ১ . . . , ৯ এর জায়গায় ০, ১। কিন্তু যেই বদলালাম, গুণ করে একক থেকে দশক থেকে শতক এগিয়ে যাওয়ার কাঠামোটা একই রইল। শুধু গুণকের মানটা এবার আর ১০ নয়, ২। কারণ সংখ্যা বাড়ানো যায় ০ থেকে ১। তারপরে আর অঙ্ক নেই। তাই যাও পরের ঘরে। আগের বার যেটা বাড়ত ১০ করে এখন সেটা ২ করে বাড়ছে।

তার মানে, এবার এই গুণকগুলোর একটা তালিকা বানানো যাক। দশমিক বা বাইনারি পদ্ধতির গুণক আর বাইনারি বা দ্বিত্ব পদ্ধতির গুণক। এই তালিকাটা বানানো বাঁ থেকে ডানে — কম থেকে বেশি। আর প্রতিটি গুণকের নিচের ঘরে আমরা তার আঙ্কিক মান দিয়েছি। দুটো মানই দিয়েছি ডেসিমাল নিয়মে। শুধু একদম নিচে একটা সম্পূর্ণ অপ্রয়োজনীয় লাইন যোগ করেছি যেটায় বাইনারি গুণক গুলোর বাইনারি মান। এবং দেখুন তিন আর ছয় নম্বর লাইনদুটো হুবহু এক। এটা সংখ্যা পদ্ধতির আভ্যন্তরীন সৌষাম্য। এমনকি যদি এখানে ডেসিমাল আর বাইনারির জায়গায় হেক্সাডেসিমাল বা অক্টাল মানে ১৬ বা ৮ অঙ্কের নিয়ম আনতাম তাহলেও একই ঘটত। যাকগে, এবার আমাদের ঠাকুর্দার বয়সের ম্যাজিকের মজায় ফেরত আসার সময় হয়েছে।

পদ্ধতি	১-নম্বর ঘর	২-নম্বর ঘর	৩-নম্বর ঘর	৪-নম্বর ঘর	৫-নম্বর ঘর	৬-নম্বর ঘর	৭-নম্বর ঘর	৮-নম্বর ঘর	. . .
ডেসিমাল গুণক	$১০^{০}$	$১০^{১}$	$১০^{২}$	$১০^{৩}$	$১০^{৪}$	$১০^{৫}$	$১০^{৬}$	$১০^{৭}$	. . .
ডেসিমাল গুণকের ডেসিমাল মান	১	১০	১০০	১০০০	১০০০০	১০০০০০	১০০০০০০	১০০০০০০০	. . .
বাইনারি গুণক	$২^{০}$	$২^{১}$	$২^{২}$	$২^{৩}$	$২^{৪}$	$২^{৫}$	$২^{৬}$	$২^{৭}$	. . .
বাইনারি গুণকের ডেসিমাল মান	১	২	৪	৮	১৬	৩২	৬৪	১২৮	. . .
বাইনারি গুণকের বাইনারি মান	১	১০	১০০	১০০০	১০০০০	১০০০০০	১০০০০০০	১০০০০০০০	. . .

০.৮.২.২।। ফেরত এলাম ম্যাজিকে

এবার ঠাকুর্দার উত্তরমালার ওই সংখ্যাটা মনে করুন — ১১০১১০১। এটা একটা বাইনারি সংখ্যা যদি হয়, তাহলে এর মান কত হবে?

পুরো ফর্মুলা ভাই — বসানো যাক। শুধু এখন ডেসিমাল গুণকের জায়গায় বাইনারি গুণক।

$$
\begin{aligned}
১১০১১০১ &= ((১ \times ২^{৬}) + (১ \times ২^{৫}) + (০ \times ২^{৪}) + (১ \times ২^{৩}) + (১ \times ২^{২}) + (০ \times ২^{১}) + (১ \times ২^{০})) \\
&= ((১ \times ৬৪) + (১ \times ৩২) + (০ \times ১৬) + (১ \times ৮) + (১ \times ৪) + (০ \times ২) + (১ \times ১)) \\
&= (৬৪ + ৩২ + ০ + ৮ + ৪ + ০ + ১) \\
&= ১০৯
\end{aligned}
$$

পরখ করে দেখুন, এই প্রক্রিয়ায় যে কারুর বয়স আপনি হিশেব করতে পারবেন। এটা দেখে মজা লাগছে, কিন্তু প্রোগ্রামিং-এর সঙ্গে প্রাথমিকতম পরিচয় যাদের আছে তাদের কাছে স্পষ্ট — এটাই বাইনারি ডিভিশন অ্যালগরিদম। ভেবে দেখুন কারুর বয়স যদি ১২৮ এর কম, মানে ০ থেকে ১২৭ এর মধ্যে হয়, তাহলে সবচেয়ে বেশিবার জিগেশ করতে হলেও সেই সংখ্যাটা হবে ৭। সবগুলোরই উত্তর হ্যাঁ হলে সংখ্যাটা হত ১১১১১১১ মানে ডেসিমাল নিয়মে ১২৭। আর ধরুন আপনার যদি কোনো বাচ্চা এখনো হয়নি, তার বয়স ০০০০০০০ মানে ০। কারুর বয়স ৯৬ হলে হত ১১০০০০০। মানে প্রথম দুটোয় হ্যাঁ, পরের সবকটাই না। এইভাবে যেকোনো সংখ্যাই মিলবে। এভাবেই আসে আঙ্কিক তথ্যের বাইনারি আকার।

০.৮.২.৩।। আঙ্কিক তথ্যের বাইনারি আকার

আমরা আগেই বলেছি একটা জটিল প্রশ্নকে ভেঙে নেওয়া টুকরো টুকরো সরল প্রশ্নে — যার প্রত্যেকটারই উত্তর হ্যাঁ বা না, মানে ১ বা ০। এই প্রশ্নমালাটা এবার পরপর উত্তর দিয়ে গেলেই আমরা তথ্যের বাইনারি আকারটা পাচ্ছি। যেখানে প্রত্যেকটা উত্তরের একটা অবস্থানগত মান এবং একটা গাণিতিক মান আছে। যেমন প্রথম প্রশ্নটার উত্তরের অবস্থানগত মান ৭, গাণিতিক মান ৬৪, পরের প্রশ্নের অবস্থানগত মান ৬, গাণিতিক মান ৩২।

এই সমস্যাটাকে এবার নিছক গাণিতিক ভাবে ভাবুন। মোট কতগুলো আলাদা আলাদা বয়স হতে পারে? শুধু পূর্ণসংখ্যার নিরিখে সম্ভাব্য সমস্ত আলাদা আলাদা বয়সের সংখ্যা ১২৮ — মানে ০ থেকে ১২৭। এই ১২৮ টা আলাদা আলাদা উত্তরকে পরপর লিখে যেতে হলে আমাদের লিখতে হত ০০০ থেকে ১২৮। এই ১২৮-টা আলাদা আলাদা সম্ভাবনাকে আমরা হাজির করে দিতে পারছি একটা সাত অঙ্কের বাইনারি সংখ্যায় — তার অবস্থান মানটা বদলে বদলে। একটা গাণিতিক তথ্যের বেলায় এটা সবসময়েই সত্যি — একটা সংখ্যায় কতগুলো আলাদা আলাদা তথ্য আছে সেটা আমরা পাব সেই সংখ্যাটাকে বাইনারিতে লিখতে যে কটা অঙ্ক লাগে তার সংখ্যা দিয়ে — অবস্থান মান যেখানে হতে পারে ০ বা ১। এই ক্ষেত্রে সেটা ৭।

এবার একটা ছোট্ট তালিকা বানিয়ে ফেলা যাক, কত বিট তথ্য দিয়ে আমরা কত বড় ঘটনাকে বোঝাতে পারব। এক বিট মানে ধরুন একটা খোপ — যেখানে রাখা যায় হয় একটা ০, নয় একটা ১। এই তালিকাটা আমাদের পরের দিনের আলোচনাগুলোতেও কাজে লাগবে।

বিটের সংখ্যা	কতগুলো সম্ভাব্য ঘটনাকে হাজির করা যায়
১	২
২	৪
৩	৮
৪	১৬
৫	৩২
৬	৬৪
৭	১২৮
৮	২৫৬
৯	৫১২
১০	১০২৪

...	...
১৬	৬৫৫৩৬
...	...
৩২	৪২৯৪৯৬৭২৯৬
...	...
৬৪	$১.৮৪৪৬৭৪৪০৭ \times ১০^{১৯}$

০.৮.৩।। একটা টেক্সট ফাইলে ভরে রাখা তথ্যের পরিমাপ

আমরা প্রথমেই একটা টেক্সট ফাইলকে তুলে নিলাম কারণ কম্পিউটারে কাজ করার সময় সবচেয়ে প্রাথমিক একটা কাজই থাকে একটা টেক্সট ফাইলকে বদলানো বা এডিটিং। এই ফাইলটা সে অর্থে বিশুদ্ধ টেক্সট ফাইল নয়, এর মধ্যে অনেক ফর্মাটিং আছে, গ্রাফিক্স আছে — মানে টেক্সট ছাড়া আরো অনেক কিছু। আগেই তো এসেছে এই আলোচনাটা।

বিশুদ্ধ একটা টেক্সট ফাইল এখানে তুলে দিচ্ছি আমরা।

```
# Sample .profile
#
# This file is read each time
  # a login shell is started.
# All other interactive shells
  # will only read .bashrc;
  # this is particularly
  # important for language
  # settings, see below.

test -z "$PROFILEREAD" && . /etc/profile

# Most applications support
  # several languages for their output.
# To make use of this feature,
  # simply uncomment one of the
  # lines below or
  # add your own one
  # (see /usr/share/locale/locale.alias
  # for more codes)

#export LANG=de_DE@euro  # uncomment for German
#export LANG=fr_FR@euro  # uncomment for French
#export LANG=es_ES@euro  # uncomment for Spanish

# Some people don't like fortune.
# If you uncomment the following lines,
  # you will have a fortune cookie
  # each time you log in ;-)

if [ -x /usr/bin/fortune ] ; then
    echo
    /usr/bin/fortune
    echo
fi
```

এই ফাইলটা কী, তা নিয়ে ভাববেন না এখন, প্লিজ। পরে আসবে সেসব বিতিকিচ্ছিরি ব্যাপার। ফাইলটার নাম '.profile' — গ্নু-লিনাক্সের ভারি জরুরি একটা ফাইল, খেয়াল করুন, নামের গোড়াতেই একটা '.' আছে, এদের আমরা ডটানন ফাইল বলে ডাকব পরে, নামের গোড়ায় একটা বিন্দু বা ডট থাকায়, এরা হল গ্নু-লিনাক্সের সিস্টেম ফাইল। এই ফাইলটাতে কোনো ফর্মাটিং নেই, ছবি নেই, বিশুদ্ধ কিছু ইংরিজি অক্ষর আছে। এটা বাংলায় করা

যেতনা, কারণ, এখনো অন্তত যেভাবে বাংলা অক্ষর আমরা ব্যবহার করছি — সেটা সরাসরি কম্পিউটারে যায়না, যায় একটা ছাঁকনি বা ফিল্টারের মধ্যে দিয়ে। লিনাক্সে অঙ্কুর বাংলায় সায়মিন্দুরা করছে, কাজ কিছুটা এগিয়েছেও যেখানে কম্পিউটারে তথ্যের স্তরেই সরাসরি বাংলাকে দেওয়া যাবে হয়ত। কিন্তু এই যে ফাইলটা আমি লিখছি এটা আমার চোখের সামনে স্ক্রিনে মেশিন বাংলা অক্ষর ফুটিয়ে তুলছে, বা প্রিন্টারেও পাঠাচ্ছে — কিন্তু সেটা ওই ফিল্টার করে। যে মেশিনে এই অক্ষর নেই আলাদা করে ইনস্টল করা নেই, সেখানে এটা পড়া যাবেনা। কিছু ল্যাটিন বর্ণমালা আর কিছু কম্পিউটারের নিজের অ্যাসকি চিহ্ন ফুটে উঠবে। সেই মেশিনে এই লেখাটা পড়ার জন্যে পিডিএফ করা হচ্ছে (PDF — **P**ortable-**D**ocument-**F**ormat) — যা সমস্ত অক্ষরগুলোকে ধরুন একটা ছবির মত করে নিজের মধ্যে রেখে দেবে — যাতে যেকোনো মেশিনে সেটা পড়া বা ছাপা যায়।

যাইহোক, এবার এই ফাইলটাকে ভাবুন। এটা আর আগের সেকশনের মত গাণিতিক তথ্য নয়। শুধু সংখ্যা দিয়ে ভরা নেই ফাইলটা। আছে শব্দ, শব্দগুলো তৈরি হয়েছে অক্ষর দিয়ে। শব্দগুলোর মধ্যে মধ্যে ফাঁকা জায়গা বা স্পেস এবং যতিচিহ্ন। এবং কয়েকটা করে শব্দ মিলে এক একটা লাইন। তার মানে ফাইলের মধ্যে কোথায় লাইন ভাঙতে হবে সেই অদৃশ্য যতিচিহ্নটাও আছে।

এই ফাইলটায় মোট ৩৫ লাইন, ১৪০ শব্দ, এবং ৮৯৯ অক্ষর। এই অক্ষর মানে ইংরিজিতে ক্যারেকটার, যার মধ্যে স্পেস আছে। আছে নিউলাইন মানে লাইন ভেঙে নতুন লাইন শুরু করার আদেশসূচক চিহ্ন। এইরকম একটা বিদঘুটে ফাইল বলে এখানে শব্দ বলতে আমরা যা বুঝি তার ধারণাটাও বদলে যাচ্ছে। যেমন, দ্বিতীয় লাইনে ওই # চিহ্নটাও একটা শব্দ বলে বিবেচিত হচ্ছে। ফাইলটার বাইটের সাইজও ঠিক ৮৯৯, একটা কোনো স্পেস কোথাও লাগালেও অক্ষরের সংখ্যা বাড়ছে ১ করে এবং বাইটও বাড়ছে ১ করে। একই জিনিষ ঘটছে কোথাও এন্টার মেরে নতুন লাইন শুরু করলেও।

এবার ভাবুন একটা এই ধরনের ডকুমেন্ট বা লেখার ফাইলে কতটা তথ্য থাকে তার সম্ভাব্য হিশেবটা। ধরুন আপনি ব্যবহৃত বা ব্যবহারযোগ্য সম্ভাব্য সমস্ত চিহ্নগুলোকে একটা তালিকা করছেন। এবার ওই তালিকা অনুযায়ী প্রতিটি অক্ষর মানে ক্যারেকটারকে এক একটা আলাদা সংখ্যা দিয়ে চিহ্নিত করছেন। এবার আবার সেই পুরোনো কায়দাটা ব্যবহার করুন — ঠিক কতগুলো সরল প্রশ্নে (মানে হ্যাঁ বা না উত্তর দেওয়া যায় যাদের) একটা ডকুমেন্টের মোট তথ্যকে হাজির করতে পারবেন আপনি আপনার এই তালিকাটা কাজে লাগিয়ে?

ধরুন একটা ৩৫০ চিহ্নের ডকুমেন্টে আপনি মোট বাহান্নটা আলাদা আলাদা চিহ্ন পেলেন। তার মানে আপনার তৈরি চিহ্ন-তালিকাটার সভ্যসংখ্যা হল বাহান্ন। ০ থেকে ৫১ অব্দি বাহান্নটা সংখ্যা দিয়ে এদের আপনি চিহ্নিত করলেন। এবার ওই ডকুমেন্টের যে কোনো একটা চিহ্নকে চিনে নেওয়া যাবে তালিকার একটা নির্দিষ্ট সংখ্যা দিয়ে। এই সংখ্যাটাকে ধরুন আমরা বললাম সূচক-সংখ্যা। এবার ওই চিহ্নটা — যা কোনোভাবেই একটা গাণিতিক সত্তা নয় — তাকে এবার বুঝছি ওই সূচক-সংখ্যাটা দিয়ে। তার মানে এবার ওই অগাণিতিক চিহ্নটারও একটা সাইজ পেয়ে গেলাম আমরা — ওই সূচক-সংখ্যাটার সাইজ। কারণ, এই সূচক-সংখ্যাটা পেলেই আমাদের তালিকা থেকে আমরা চিনে নিতে পারব ওই চিহ্নটাকে। তার মানে এবার গোটা ডকুমেন্টটার সাইজ কী দাঁড়াল? যতগুলো চিহ্ন এসেছে ডকুমেন্টটায় তাদের প্রত্যেকের সূচক-সংখ্যার সাইজ গুণ যতবার করে তারা প্রত্যেকটা এসেছে। মানে, একটা সম্পূর্ণ অগাণিতিক ফাইলেরও একটা গাণিতিক পরিমাপ আমরা পেয়ে গেলাম।

কিন্তু এই পরিমাপটা কোনো অর্থ তৈরি করতে পারবে, বা সেই ফাইলটা থেকে আমরা ডকুমেন্টটাকে বানিয়ে ফেলতে পারব শুধু ওই প্রথমেই বানিয়ে ফেলা তালিকাটার সাপেক্ষে। ওটা না-থাকলে আমরা বিশ বাঁও জলে। পরে আমরা দেখব এই তালিকাটার কিছু সর্বজনগ্রাহ্য মানদণ্ড বা স্ট্যান্ডার্ড আছে — যেমন অ্যাস্কি বা এক অর্থে ইউনিকোড। যে তালিকাটা সবাই মানে। এতে সুবিধেটা এই যে একটা তালিকা দিয়েই সবার কাজ চলে যায়। কেউ আপনাকে এবার একটা ফাইল পাঠাল — যাতে পরপর শুধু সংখ্যা, আর বলে দিল যে এটা টেক্সট ফাইল। আপনি তালিকাটা বসালেন, ঘ্যাঁচাঘ্যাঁচ পড়ে ফেললেন। এই প্রক্রিয়াটার মূল প্যাঁচটা এইখানে — চিহ্নকে সংখ্যা করে নেওয়া। এখানে মূল ডকুমেন্টটাও, তার ইলেকট্রনিক আকারটাও বদলে যাচ্ছে। যা ছিল একটা চিহ্নের সমাহার, হয়ে দাঁড়াচ্ছে একটা সংখ্যার সমাহার। আমাদের উদাহরণটা ভাবুন — ০ থেকে ৫১ অব্দি বাহান্নটা আলাদা আলাদা চিহ্ন

আছে। এবার আগের পাতায় দেওয়া কতগুলো বিট দিয়ে কতগুলো সম্ভাব্য ঘটনাকে হাজির করা যায় তার তালিকাটা দেখুন। এখানে ৫-টা বিট ম্যাক্সিমাম ধরতে পারে ৩২-টা ঘটনা, আর ৬-টা বিট পারে ৬৪-টা ঘটনা। তার মানে আমাদের ৬-টা বিট লাগবে প্রত্যেকটা চিহ্নকে বোঝাতে। ৬-বিটের একটা ছক মোট ৬৪-টা আলাদা আলাদা চিহ্নকে হাজির করতে পারবে। বরং আমাদের বারোটা রয়ে যাবে অব্যবহৃত। এবার এই ডকুমেন্টটার আকার কী দাঁড়াবে? মোট যত চিহ্নের ডকুমেন্ট — এই উদাহরণে সেটা ৩৫০ — তাকে গুণ করে দিন ৬ দিয়ে। পেয়ে গেলেন আপনার ফাইলে মোট বিটের সংখ্যা।

এই নিচের অংশটাকে একটা ভিশুয়াল চ্যাংড়ামো বলতে পারেন। আপনি কি আন্দাজ করতে পারছেন — এই বস্তুটা কী? পারা সম্ভব না — এই যে পিডিএফ ফাইলটা আপনি পড়ছেন — এই ফাইলটাকেই কোনো এক্সপিডিএফ বা জিজিভি বা অ্যাক্রোরিড


```
‰dRiTS‰•¾/»[Ü&&ú^ÜPQYÜ•˘7D@•TF©Šb˚!$†$d´l˚a ‹ƒ¯•
!,IØ·!`˙•´‘º Öe:.c Ö©3=§Óö´••:§•ÿöÜ?wýîù¾{••WÌ=ñ|ûÆì••7.•é]2K\Úµ*!t•,B yöæÀàÃ‘q¦N±DÁËTrD|•
£àÉ>úR•L!àˆR˘iireF˙G! ˆ¥2Iª’« qR˜\•ˆ#õÆW>6á‰••‘DÅ•óyÞ´ˆçÇþ, Ç-
dH•^@v„˜/•óx2 ˜ïMúoe¨…„ìØ•²ÓŸ•þqe*;%‡ý‘û•'Ç?N"Ëàˆþ •^˜"ˆT t …K“À•`• -
ÀV˘¯•`'Àn"ï˜8 @.Àa #˚DÒà$ˆŸ´$ˆœ˜H"Àm »-”ˆ4ˆRF¯= •@`ˆ¤,••`••`ˆ_!H•À- q¯&´ª´òHPLD¶°a-
â‹lY•˙¨H••Ñ•Cˆ'$Àiâ9¸Hä"i4)"‡,È´5¡„p•¨'Õ•êlV°#mó˙¼ønÂM•C& ;¤¨¤iÒsøy˘ßc`÷~• ¼zÏ•V•þI¼KÚDJ"
e’*ÉçÈ˙ò{J,ÅJ]@UQ?Ð¢i÷çEÍß:ÿò,ý£•lè¨é®Å¶%ç-ú--YújÙñe¯-•>            úd’‘Èøe…aÅô§f-
¦OÏJîÊ9fëèªÕ«^®nD×£9˙¨»Ë®^#]3¾vd e=º¾fÃÅ?ú¶l”lÚô³ßˆ¿ oqo½µ-Ä »¿Éÿq”-j}•v*¬-
ì/ûwî•îŸ=•~ÀvpÍÁöCìC=‡×•®•
w•¹•±,¢"âÅÑüHzdNä»¨ãQ=Ñ•¢s¢g  86•Ã‹•ˆ]•kŠýåxA•Ä%Æý#^•_êl
{*èçÔ£_¿-y÷r¼×ùp•±´Náó] ï$Î,•10æ,)®42^'¢\Ô˘ð7Ä
'úÜ-™ä¹xÑEiz´M[V\ a•?ðçç¯'-º3[„R¥L©´i
úªŠŠJTVø›ŒS¬d¡Ý\¥7•®av“Ën6ÑÛ†ðƒÌ›O•¹…åºÒb¬C«®ÎaÉs•òìfE««{äíÖ©D¬•O¢œd>åiyÊ¸g^F¸N
Ê:"µö×éÍ&•¦¬'kótJ53îÆ•üÓk¨õ¥³¾‡5f×H‹ÊòÕ…˜g£ÇÍ•¹¢ÆÄuÃ •ÚÝ˙š¯öRÞ•”•ÿÍ}ÿÉ¨Ýmt[ÜÓN¤©íQ-
^Ý‰Çvº¾Î&¦3ZsÉy7¯ßf~g ðà••3MÎJ’æg².œé}˜,‰N‹’Y§ùÝí¥XMUuµ±’63BIw’ÿû˜¥ß«U<…ÏË<-
b¦ô ¼í´ZZúÑú¢:u.3=U•ÍÍhîÈ@¥m¼•¥oeM>•”ÊŸ=” ±Ü÷zÆ*±Š‡ØýâücÖã¯G¦GR&…V,%A®Q°”y¦ÆÖ
•‰{˜Xjë¤à            «®±Jß,ñbÏÄ2 Z.4&¢•ûx,CnË•Ð\/ê“º…í4N’@ ÍÊÕ˜•-]¿ýÆ  ê½-1Å26^Ó×c•ådE-
V"d••J-q¶5·
 Z£JÃ¼|‘•ÃOkè•£ô7ÝÓ ÅIÚ¾•B~šø©•?5DÄ?ÌìfÜ” ½qœ•-tö´°AeÖ×TV×bVw³™ÕTU¬½’œ•/Á&/“'˙šz{˜
E"GQ¦S£M_8•cc¼c-¨YV_PÙdh®6-¬2•»
Öº•Kvu´Í¤2fª™ª¢lU>jÒ^1ä²ø9ò
A‡¼Ïj«©1b†¼u´“ËáGx˜{_àõëP¹‰ìù#Nx¿•÷c½•½ó¨IëÐ6cÍuÖ>¨Ó‘×‘|•Uvfºòmj“Oqr´O#§Ñß•§G¦:Gñåã
ÝN¯×
â”fÇ•_†û0”J“Ä*iÎ^©j´••XuF³˘³Ùû»ºZh-
¦&ƒ™emÎOéÀ,©ƒ©ä•ë·T¾“<÷Œ2è½ø51ålq”ÔC•»‘è3U2¬üò²]åäÕˆró”*¥šVRœ§+f•)ÌC¹•¾ÙÂÀK¨CON
\Ð••-•cW) ¥Š†<V-J••Õïìoq
? -‡uã¡sÄ2ê$Wi“²²äyr•ÆÅ••¨r* ,Åq´ãþÐWVg• îy¬³âOÇ¿w"“ÃD ü
#•ú9¾ö} ¾²\w-J-÷å2w•µµ˘¨EcÉµhh6ª¶]7ú’y½òY•Í-Ïi-©¡ÕèŠªJY•ÏÊàÏ±˙<” _ýõ§=••ºÇÕü
Á-••~Œ-
Bòí¦Ñ‰QæÔ…^±Bš'ÌlUYkëôF#jÔêµL¹˜•/Bé•Óx‘•Á‹q¨ÃSI•¹ä•éWÓ¨¤ÿGbÿ•#Ó•ÕÖa³µ·gÙ¤
2eV&ú?
ËÄ©ÉüŒã Ù•ÌnéLg_¨ß·}ß2mw×£ëxLµ²êîV•TPÀ#á¾r „œ@B¸Ñ@„„„œ$„$ ‹@˚WŒ´TDñØˆ
v•êº»•Övq- ]Ù_¦o;ÓØ?à™ù>ßg•ï÷“¨Ùˆ,œÉ›ñw¯o 3¸<qy˜ø"Æ„~$gxC%´Äksø{;¬í*ì59•˚ÿi'ûÊ
H>²»Ó-
ÒVQiõˆ˙®•âxGF¬s¨R©íÒn¨¯âLÊáB~ Î×Œµ́µµ·´ROi=˘ÓˆÃ¦ýk•³9Ãc® î¶¤ŠòüBø¢eŸ‘•
HõÁÆofà•Ï½ÇKèÉ«Ós•,î-•.’²¯[o…=Ìèv´ÕÜdÒRÁæÍ ‘ÜÝ®hk…yˆ•-]:<Ü•ßåå$ÐL•nG ¶˜
·KiÂð=”l»Èî6Z¬•‰YÐÖÜÑÞ,æƒ
```


জাতীয় কোনো পিডিএফ ভিউয়ার দিয়ে না-খুলে খোলা হয়েছে একটা বিশুদ্ধ টেক্সট পড়ার সফটওয়ার দিয়ে — ফর্মাটিং ছাড়া টেক্সট যাতে পড়া যায়। ফাইলটার একদম শুরুটা এখানে তুলে দিলাম। লাইনভাঙাটা একটু বদলালাম, জায়গা বানাতে। পিডিএফ ভিউয়ার সফটওয়ারগুলো এই ফাইলটা পড়ে এবং স্ক্রিনে বা প্রিন্টারে পাঠায় ফর্মাটিং-সহ ওই টেক্সট যা এতক্ষণ আমরা পড়তে পড়তে এলাম।

এবার আমাদের দেওয়া ‘.profile’ ফাইলটার এই মাপগুলোর সঙ্গে আলোচনাটাকে মিলিয়ে নিন। শুধু ওই ক্ষেত্রে একটা জিনিষ মাথায় রাখবেন — আট বিটে এক বাইট। আর এক এক বাইট সেখানে এক একটা চিহ্নকে হাজির করছিল। কেন — সেই আলোচনায় আমরা এখুনি যাচ্ছি।

০.৯।। বিট বাইট ওয়ার্ড — স্মৃতির একক

ধরুন একটু আগের আমাদের উদাহরণের কাল্পনিক ডকুমেন্টটাকে। যাতে মোট চিহ্ন ছিল ৩৫০-টা, ৫২-টা আলাদা আলাদা চিহ্নের সমাহার। ধরুন এই ডকুমেন্টটা লাবণ্য পাঠাল অমিত-কে। এবং আরো দেখা গেল যে লাবণ্য আর অমিত-র (বস নিজে 'অমিতের' লেখা পছন্দ করতেন না, লেখা চাই 'অমিত-র') মধ্যে ডকুমেন্টগুলোতে ওই বাহান্নপিস চিহ্নই থাকে। এবার লাবণ্য আর অমিত তাদের সরোবরের দুপাশে দুজনের প্রাসাদের চূড়োয় ছপিস করে লালবাতি লাগাল। (লালবাতিটা আমি না বস নিজেই বলে গেছেন। রঞ্জিৎ সিং বলেছিলেন না, হিন্দুস্থান একদিন লালমে-লাল হয়ে যাবে — তারই বাংলা ভার্শন হয়ত।) এবার জ্বলে থাকা মানে ১ আর নিভে থাকা মানে ০। বেড়ে এখন চিঠিচালাচালি চলছে লাবণ্য আর অমিত-র মধ্যে। এবার প্রশ্ন উঠছে হারাধনের ওই বারোটি উপরি ছেলে — তারা কি কিছুই করবেনা — পড়ে থাকা ছ-বিটের ওই বারোখানা সমাহার। করবেনা আবার কী — আলবত করবে — রিতুর পর্নো থেকে, একটু সফট যদিও, আপনি হদিশ পেতে পারেন — আর কী কী পোয়াবারো চলবে, মানে চলতেই পারে ওই পড়ে থাকা বারোয়। মজার কথা কী বলুন তো — ঠিক এটাই ঘটেছিল টেলিটাইপরাইটারের বেলায়। পড়ে পাওয়া চোদ্দআনার মত এই ফাউ বিটসমাহারগুলোয় কিছু টেক্সটোত্তর চিহ্নকে বুনে দেওয়া হয়েছিল যা যাবে টেক্সট-এর সঙ্গেই।

ইংরিজি ভাষা ব্যবহার করে ল্যাটিন বর্ণমালা — যাতে ছাব্বিশখানা অক্ষর। আর প্রত্যেকটারই দুটো করে হাত আছে — বড় হাত আর ছোট হাত। মানে বাহান্ন। দশখানা অঙ্ক মানে ০ থেকে ৯। হল বাষট্টি। নয় নয় করেও কয়েকপিস যতিচিহ্ন। পিরিয়ড বা ফুলস্টপ, কমা, সেমিকোলোন, কোলোন, যোগ চিহ্ন, বিয়োগ চিহ্ন, অ্যাপোস্ট্রফি বা উর্দ্ধকমা, এবং কোটেশন মার্ক ইত্যাদি। মোট এতেই দাঁড়াল সত্তরখানা। কিন্তু কিন্তু একদম গোড়ার দিকের টেলিটাইপরাইটার ব্যবহার করত পাঁচ বা ছয় বিটের কোড। মানে ৩২ বা ৬৪-টা আলাদা আলাদা সম্ভাব্য চিহ্ন। তখন এল সাত বিটের ব্যবস্থা। মানে ১২৮-টা আলাদা আলাদা চিহ্নের সম্ভাব্যতা। যাতে এই সবগুলো তো দেওয়াই যাচ্ছে, যাচ্ছে আরো কিছু। ফাউ। তখন যে সর্বজনগ্রাহ্য মানদণ্ডটা তৈরি হল সেটাই অ্যাসকি (ASCII — **A**merican-**S**tandard-**C**ode-for-**I**nformation-**I**nterchange)। এতে সাত বিটের ব্যবস্থা। এই ১২৮-টা সম্ভাব্য চিহ্নের মধ্যে ৯৬-খানায় এল সেই সব চিহ্ন — যারা ছাপা হবে, প্রিন্টেবল ক্যারেকটারস। মানে অক্ষর, সংখ্যা, যতিচিহ্ন ইত্যাদি। অন্য ৩২-টা হল সেই টেক্সটোত্তর চিহ্নরা। এদের বলে কন্ট্রোল ক্যারেকটারস। এদের মধ্যে আছে ক্যারেজ রিটার্ন — এই নামটাও আমরা আজো ব্যবহার করছি। কিন্তু আসলে এটা আসছে সেই ম্যানুয়াল টাইপরাইটারের সময় থেকে — যখন ক্যারেজটা তার বেঁকা হাতল সহ ফিরিয়ে নিয়ে যেতে হত টাইপ করার সময় নতুন লাইনে যেতে হলে। আছে লাইন ফিড — মানে পাতায় এক লাইন নেমে এসো। এবং আছে সেই বাঁশরীধ্বনি। কোড ৭। যার প্রসঙ্গ থেকেই এই আলোচনার শুরু। টেলিটাইপরাইটারে ঘন্টা বেজে ওঠার আদেশ। কারণ লোককে তো টের পেতে হবে যে বার্তা এল। মজার কথা জাক লাকাঁর সেই লুপ্ত মেটাফরের মত আজো সেই একই কমান্ড তার অ্যাসকি কোড সহ ব্যবহার হয়ে চলেছে (আমরাও করব, ব্যাশ স্ক্রিপ্ট নিয়ে আলোচনায়। দশ আর এগারো নম্বর দিনে), যদিও আজ আর ৭-বিট কোড নয়, এখন ন্যূনতম এককটা আট বিট, মানে এক বাইট।

১৯৬০-এ, আইবিএম ৩৬০ মেশিনের সময় থেকে এল এই ৮-বিটের কোড। এর মূল কারণ ছিল, এক, এতে ঠিক দ্বিগুণ সংখ্যক নতুন চিহ্ন তারা আনতে পারলেন। আর ক্রমশ বাণিজ্যের তথা ইলেকট্রনিক যোগাযোগের জগতটা এত দ্রুত বেড়ে চলেছিল যে রোজই নতুন নতুন চিহ্নের প্রয়োজনও বাড়ছিল। আর, দুই, এতে একটা তথ্য সমাহারের মধ্যে বিট-গুলোকে আরো দক্ষভাবে ব্যবহার করা যাচ্ছিল।

এবার আমরা আরো অনেক ভালো করে বুঝতে পারব এই লেখাতেই ০.৮.৩ নম্বর সেকশনে আমরা বিশুদ্ধ টেক্সট বলতে কী বুঝিয়েছি। যেসব ডকুমেন্ট ফাইলে ব্যবহৃত চিহ্নগুলো ০ থেকে ১২৭-এর মধ্যে থাকছে তারাই বিশুদ্ধ টেক্সট ফাইল। ১২৮ থেকে ২৫৫ অব্দি চিহ্নগুলো মানে এক্সটেন্ডেড অ্যাসকি কোডকে কাজে লাগিয়ে যে জটিলতর ফর্মাটিং সেগুলো ওইসব ফাইলে নেই।

যেমন বললাম, স্মৃতির ন্যূনতম একক এখন এই আট বিট। যাকে ডাকা হয় এক বাইট বলে। কিন্তু দ্রুতগতির পিসি আর অপারেটিং সিস্টেমরা যখন তাদের কম্পিউটারের স্মৃতি (মেমরি এবং ভাঁড়ার দুই-ই) ব্যবহার করে তারা একটা

বাইটের এককে করেনা, করে ২, ৪, বা ৮ বাইটের এককে। তখন সিস্টেম দাঁড়ায় ১৬ বিট, ৩২ বিট বা ৬৪ বিটের। এবার দেখুন তো, এই লেখাটার ১০ নম্বর পাতায়, এক্সপ্যানশন স্লটের ছবিতে গোটাটা আপনি আন্দাজ করতে পারছেন কিনা? স্মৃতিকে ব্যবহার করার এই ২, ৪ বা ৮ বাইটের, মানে ১৬, ৩২ বা ৬৪ বিটের একককেই তখন আমরা ডাকি ওয়ার্ড বলে। কম্পিউটারের ভাষাচর্চার একক। এই নিরিখেই সে লেখাপড়া করে। মানে স্মৃতিকে লেখে র‍্যামে রমে বা ডিস্কে, এবং প্রয়োজন মোতাবেক সেখান থেকে পড়ে।

অ্যাসকি চিহ্ন-তালিকা।। অ্যামেরিকান স্ট্যান্ডার্ড কোড ফর ইনফরমেশন ইন্টারচেঞ্জ-এর এই তালিকাটা তৈরি করা সাত-বিট কাঠামোয়। এর পরে তৈরি অনেক আট-বিট তালিকা (যেমন আইএসও-৮৮৫৯-১, যা গ্নু-লিনাক্সের স্বাভাবিক চিহ্ন-তালিকা) তার নিচের অর্ধেকে এই তালিকাটা ব্যবহার করে। অ্যাসকি ইংরিজি বর্ণমালার জন্যে তৈরি। এর আন্তর্জাতিক বিকল্পটার নাম আইএসও-৬৪৬। এটিঅ্যান্ডটি-র ইউনিক্সের ভার্শন ৭-এ অ্যাসকি ম্যানুয়াল প্রথম আসে। ১৯৬৮-তে এই অ্যাসকি চিহ্ন-তালিকা প্রকাশ করে ইউনাইটেড স্টেটস অফ আমেরিকা স্ট্যান্ডার্ডস ইনস্টিটিউট। এখানে আমরা তালিকায় চিহ্নের মানটা দিয়েছি ডেসিমাল পদ্ধতিতে। এখানে আমরা বর্ণমালার চিহ্নগুলো এবং অঙ্কের চিহ্নগুলো এবং যতিচিহ্নগুলো দেখাইনি। শুধুমাত্র কয়েকটা কন্ট্রোল ক্যারেকটার বা ফাউ। যারা টেক্সটোত্তর অর্থ বহন করে টেক্সট ফাইলের মধ্যেই। তাও সবগুলো না, কয়েকটা মাত্র। তালিকায় ০ নম্বর চিহ্নটা বোঝায় নাল বা কিছু-না। এর মানে কিন্তু ০ নয়। ০ একটা আলাদা চিহ্ন। নাল দিয়ে সচরাচর একটা স্ট্রিং বা চিহ্নমালার শেষ বোঝানো হয়। আমরা পরে দেখব এসব।

চিহ্নের মান	কিবোর্ডের চাবি	কী বোঝায়
০	<CTRL> + @	নাল বা কিছু-না — সচরাচর স্ট্রিং-এর শেষ
১	<CTRL> + A	হেডিং শুরু
২	<CTRL> + B	টেক্সট শুরু
৩	<CTRL> + C	টেক্সট শেষ
৪	<CTRL> + D	ট্রান্সমিশন শেষ
৫	<CTRL> + E	জিগেশ করো বা খোঁজো — এনকোয়ার
৬	<CTRL> + F	স্বীকার করো — অ্যাকনলেজ
৭	<CTRL> + G	বেল — ঘন্টা বাজাও
৮	<CTRL> + H	ব্যাকস্পেস — কার্সরের আগের চিহ্ন মোছো
৯	<CTRL> + I	হরাইজন্টাল ট্যাব — পাশাপাশি ডানে সরে যাও
১০	<CTRL> + J	লাইন ছাড়ো — লাইন ফিড
১১	<CTRL> + K	ভার্টিকাল ট্যাব — খাড়া নিচে নামো
১২	<CTRL> + L	ফর্ম ফিড — নতুন পাতা
১৩	<CTRL> + M	ক্যারেজ রিটার্ন বা নতুন লাইন
১২৭	<ALT> + 127	ডিলিট — কার্সরের পরের চিহ্ন মোছো

কমান্ডগুলো কী করে পড়ার সময় মাথায় রাখুন এখানে একটা টেলিটাইপরাইটারকে ভাবছি আমরা। একটা বার্তার ট্রান্সমিশন যা একটা টেলিটাইপরাইটার মেশিনকে খুঁজছে। ঘন্টা বাজানোর চিহ্নটা দেখুন, তাহলেই ব্যাপারটা সহজ হবে। বার্তাটা মেশিনে পৌছনোর পর চারদিকের লোকজনকে তো সচেতন করতে হবে, নতুন একটা বার্তা পৌছলো। সেইখান থেকে ঘন্টাটার প্রয়োজন ছিল ন্যূনতম বার্তার কোডেও। এই দশমিক সংখ্যা '০৭' এই কোডটা পরেও রয়ে গেছিল, আমরা পরে ব্যাশ শেল করতে গিয়ে দেখব, এই সংখ্যা দিয়েই সেখানে ঘন্টা বাজানো যায়, এখনো। যখন সেই অর্থে ঘন্টাটা আর দরকার নেই, কারণ ঘন্টার ইতিহাসটাই মরে গেছে। এই কোডটা একটা লুপ্ত মেটাফর, মরে যাওয়া ইতিহাসটা না-জানলে যার অর্থ কিছুতেই উদ্ধার করা যায়না।

এখানে এই অ্যাসকি তালিকার একটা জিনিষ নিয়ে আমাদের প্রশ্ন উঠতে পারে, পরেও আবার আসবে প্রসঙ্গটা, সেটা একটু স্পষ্ট করে নেওয়া যাক। তবে এই একদম প্রাথমিক স্তরের পাঠক এই প্রসঙ্গটা বুঝতে না-ও পারেন, তা নিয়ে চিন্তিত হবেন না, পরে ঠিক বুঝে যাবেন। এই শেষ প্যারা কটা অনেকেরেই একটু কঠিন লাগতে পারে, পরের প্রয়োজনের কথা ভেবে লিখে রাখতে হচ্ছে। এমনকি এখন এই প্রসঙ্গটা বেমালুম ছেড়ে দিয়ে পরে ফেরত আসতে পারেন, টেক্সট নিয়ে যেখানে বিশদতর আলোচনা আসবে।

এইমাত্র দেওয়া এই অ্যাসকি তালিকায় দুটো সংখ্যা দেখুন ‘১০’ আর ‘১৩’। প্রথমটা, ‘১০’, লাইনফিড, মানে লাইনছাড়ো, বা একদম কাঁচা রকমে বললে, প্রিন্টারকে বা স্ক্রিনকে আরো একটা লাইন খাওয়াও, ইংরিজিতে সচরাচর এটাকে ‘LF’ বলে উল্লেখ করা হয়। দ্বিতীয়টা, ‘১৩’, ক্যারেজ-রিটার্ন বা নতুন লাইন, ইংরিজিতে এটাকে ডাকা হয় ‘CR’। আপাতদৃষ্টিতে, দুটোকেই তো একই বলে মনে হচ্ছে, নিট কাজটা হল নতুন লাইন শুরু করা। তাহলে?

এই খ্যাঁচটা সচরাচর কম্পিউটার ক্যাওড়ামিতে ‘^M’ মিস্ট্রি বা ‘CR-LF’ ইশু বলে ডাকা হয়। সমস্যাটা আসলে দেশি-বিদেশি মিক্সড-এর। এটা গ্নু-লিনাক্স তথা যেকোনো ইউনিক্সের সঙ্গে মাইক্রোসফট ডস বা উইনডোজ জগতের একটা পারস্পরিকতার সমস্যা। শুধু যদি একটা সিস্টেমে থাকেন আপনি, সেখানের টেক্সট নিয়েই কাজ করেন, তাহলে কোনো সমস্যা নেই। সমস্যা হল একটা থেকে অন্যটায় যেতে চাইলে। আর গ্নু-লিনাক্স তথা যে কোনো ইউনিক্সকে কেন একটা ব্রাকেটে নিয়ে আসছি, বা এই বর্গটা দিয়ে আমরা ঠিক কী বোঝাচ্ছি সেটা স্পষ্ট হওয়ার জন্যে আপনাকে দিন চার আর পাঁচ অব্দি ওয়েট করতে হবে।

এখন জাস্ট এইটুকু মাথায় রাখুন, কম্পিউটার অবতার আর্বিভাবের শঙ্খরবে যখন মাতল ত্রিভুবন, সেই সময়টায়, সেটা বেশ পুরোনো সময়, খুব আলাদা, ভারত তখনো ভাঙেনি, টিভি আসেনি এই গ্রহে, দ্বিতীয় বিশ্বযুদ্ধ চলছে, আরএসএসের হাতে গান্ধী খুন হতে দু-এক বছর দেরি আছে, রবীন্দ্রসঙ্গীত নামটাই আসেনি ধরাধামে, রবিবাবুর গান তখনো এমনকি নিজের সুরে গেয়ে রেকর্ডও বার করা যেত, সেই সময় থেকেই সমস্যাটার শুরু, টেক্সট এবং সংখ্যার সম্পর্কটা ঠিক কী হবে। তারপর, কম্পিউটার ব্যবহার ক্রমে একটু একটু করে বাড়ার সঙ্গে সঙ্গে চালু হল প্রথম প্রথাটা, ইউনিক্সের পূর্বপুরুষদের হাত ধরে এল, নতুন লাইন বোঝাতে লাইনফিড বা ‘LF’। এই পাঠমালাতেই পাবেন, তখনো ডস আসতে ঢের দেরি। পরে, ডস যখন এলো, পৃথিবীর ধূর্ততম বানিয়াদের একজন বিল গেটসের হাত ধরে, তখন সেখানে চালু হল, দুটোকেই একই সঙ্গে ব্যবহারের প্রথা, মানে প্রতি লাইনের শেষেই ক্যারেজ-রিটার্ন আর লাইনফিড, মানে ‘CR-LF’।

তাই এই ডস, এবং পরে উইনডোজ প্রথায় ‘CR-LF’ এই দুটো চিহ্ন দিয়েই লাইন-বোঝানো রকমে লেখা টেক্সট গ্নু-লিনাক্স তথা যে কোনো ইউনিক্স সিস্টেমেই পড়তে গেলে, বাড়তি একটা ‘^M’ দেখা দেয়, ইম্যাক্স দিয়ে কোনো উইনডোজে লেখা ফাইল খুললেই দেখবেন, প্রতি লাইনের শেষে একটা করে ‘^M’। এর মানে ওই বাড়তি ক্যারেজ-রিটার্নটা। আবার গ্নু-লিনাক্সে লেখা টেক্সট উইনডোজে পড়তে গেলে গোটাটা ঘেঁটে যায়, লাইন ভাঙা গুলো দেখাতে পারেনা সঠিকভাবে। এই জন্যে সমস্ত ইউনিক্স বা গ্নু-লিনাক্স সিস্টেমেই নানা অ্যাপ্লিকেশন দেওয়া থাকে, আমি সচরাচর ব্যবহার করি, ‘unix2dos’ বা ‘dos2unix’, নাম থেকেই বুঝতে পারছেন, তাদের কাজ কী, শুধু ‘2’-টাকে ইংরিজি শব্দ ‘টু’ হিশেবে পড়বেন।

এই কাজটা আপনি সরাসরি নিজের হাতেও করতে পারেন, কাজটা বেশ সিম্পল। ধরুন, একটা টেক্সট ফাইল গ্নু-লিনাক্স তথা যে কোনো ইউনিক্স সিস্টেম থেকে আপনি ডসে বা উইনডোজে আনছেন। এই মাত্র আমরা দেখলাম, লাইনফিড বা ‘LF’ মানে অ্যাসকি মান ‘১০’। আর ক্যারেজ-রিটার্ন বা ‘CR’ মানে অ্যাসকি মান ‘১৩’। আর, গ্নু-লিনাক্সে যেখানে লাইনের শেষে ‘LF’ আছে, উইনডোজ প্রতিবেশে সেই ফাইল পড়তে গেলে প্রতিটি লাইনের শেষে আপনার ‘CR-LF’ মানে ‘১০’ আর ‘১৩’ দুটোকেই পেতে হবে। খুব সোজা, অ্যাসকি-তে টেক্সটা খুলে প্রতিটি ‘১০’ মানের আগে একটা করে ‘১৩’ গুঁজে দিন। ঠিক এটাই করে বদলানোর প্যাকেজগুলো। এবার বেমালুম আপনি সেটা উইনডোজ প্রতিবেশে পড়তে পারবেন।

উইনডোজ সিস্টেমে একজন সাধারণ ব্যবহারকারীর এসবের দরকার পড়ে না, কারণ, উইনডোজ কোনো গ্নু-লিনাক্স সিস্টেমকে দেখতেই পায়না। যেমন আমার মত যাদের সিস্টেমে উইনডোজ আর গ্নু-লিনাক্স দুটোই আছে, তাদের গ্নু-লিনাক্স থেকে দেখলে গোটা হার্ডডিস্ক আয়তনটা দেখা যায়, কিন্তু উইনডোজ দিয়ে দেখলে সিস্টেম দেখায় শুধু সেটুকু হার্ডডিস্ককেই যেটুকুতে উইনডোজ রয়েছে। এটা উইনডোজ সিস্টেমের একটা অক্ষমতা, পরে আমরা ভালো করে দেখব এবং বুঝব এসব। কিন্তু, তাই, গ্নু-লিনাক্স সিস্টেমকেই যেহেতু দেখতে পাওয়ার সাধ্য নেই উইনডোজের, তার ফাইলকে আলাদা করে বোঝাটা তো আরো দূরবর্তী হয়ে পড়ছে। তবে উইনডোজ সিস্টেমে বসেও যারা

সিরিয়াস কাজ করেন, তাদের জন্যে উইনডোজ প্রতিবেশে গ্নু-লিনাক্স টেক্সট ফাইল পড়ার প্যাকেজও আছে। তবে গ্নু-লিনাক্স পার্টিশনকে দেখতে পাওয়ার ক্যালি উইনডোজের এখনো হয়নি।

অনেক লম্বা হয়ে গেল এই শূন্য নম্বর দিনটা। কিন্তু কত জরুরি কিছুই তো বলা হলনা। এর মধ্যেই তথাগত অনেকগুলো খুঁত ধরে দিল। ও খুব ভালো রিডার। তার একটা সংশোধন এটাতেই করে দিলাম। আমি ভুল করে ফাইলগুলোকে এক্সিকিউশনাল লিখছিলাম, ও খেয়াল করিয়ে দিল যে এটা এক্সিকিউটেবল হবে। এটা তবু ছোটখাট। পরের অনেকগুলো খুব গুরুত্বপূর্ণ ভুল। বলছিলাম না, আমরা লিনাক্সীরা এভাবেই কাজ করি। আমাদের কমিউনিটি মিলে।

glt-mad@ilug-cal.org

MADHYAMGRAM

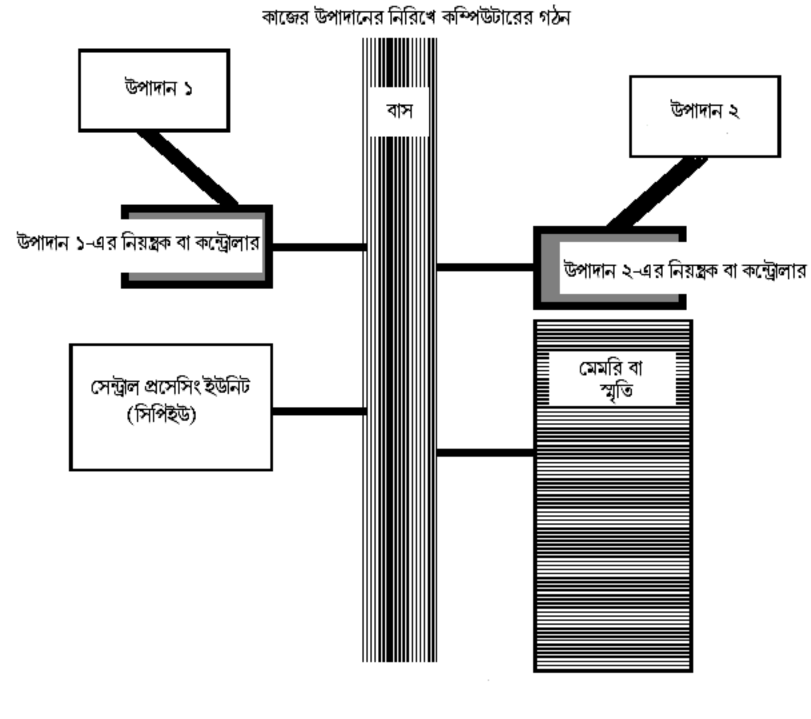
জিএলটি মধ্যমগ্রাম
glt-mad@ilug-cal.org

আমরা শূন্যতা থেকে বেরিয়ে এসেছি। এখন আমাদের সামনে পিসিটা মানে কম্পিউটারটা ইতিমধ্যেই সদাসর্বদাই, অলওয়েজ অলরেডি, আছে। আমরা এবার কম্পিউটারের কাজ করার প্রক্রিয়ার আলোচনায় ঢুকব। আজই আদতে আমাদের আলোচনার প্রথম দিন — দিন নাম্বার ওয়ান — নাম্বার ওয়ান সিরিজের প্রভু জয় গোবিন্দা জয় ডেভিড বলে শুরু করা যাক। শূন্যতম দিনে সরাসরি কোনো বইয়ের রেফারেন্স দেওয়া হয়নি, রেফারিত হওয়ার প্রয়োজন পড়বে তারই যে এই এক অব্দি আসছে। দিন শূন্য-র মূল রেফারেন্স 'পিসি হার্ডওয়ার ইন এ নাটশেল' আর পিটার নর্টনের 'ইনসাইড ইয়োর পিসি'। এর বাইরে থেকে অনেক আলোচনা এসেছে — সেগুলো বহুমুখী বিচিত্রমুখী। ছবিগুলো নামানো হয়েছে মূলত নেট থেকে, কিছু বিভিন্ন এনসাইক্লোপিডিয়া থেকে, এবং কিছু আমি, হুঁ হুঁ, সরাসরি নিজে এঁকেছি, তাও আবার বাঁ-হাতে মাউস ধরে। এবং নামানো ছবিগুলোর নকল করার অধিকার, কপিরাইট, আইন বাঁচাতে, কারণে, এবং অকারণেও, ছবির শরীরে বহু লেখা গুঁজে দেওয়া হয়েছে। অন্নু মালিকের মডেলে, এর অনু পরমানুর মালিক এখন আমরাই। এফএসএফ-এর সঙ্কর্ষণ ইতিমধ্যেই আমায় থ্রেট করেছে, দেখো, সাইটে যেন তোলা যায়। আমরা শূন্যয় কম্পিউটারের ভৌত উপাদানগুলোর কাজের ছক দেখেছিলাম। এবার শুরু করব কাজের ভিত্তিতে উপাদানগুলোর ছক দিয়ে। শুধু একটা কথা মাথায় রাখবেন, কোনো সেকশনেরই পুরোটা বোঝা যায়না অন্য সমস্ত সেকশনগুলো জেনে ফেলার আগে, আর পুরোটা বোঝা বলে কিছু হয়ই না। তাই, কোথাও, খুব ক্যাঁচাল বলে মনে হলে দাঁতে হাত রেখে পড়ে যাবেন, পরে দেখবেন এই করতে করতে কবে মাড়ি হয়ে উঠেছে শক্ততর, দাঁত পড়ে যাওয়ার কোনো ভয়ই নেই আর, বরং মনে হবে আরে, এটুকু দাঁত ও মাড়ির ব্যায়াম তো প্রত্যহ পালনীয়। আমরা একটু সিপিইউর কাজের রকম নিয়ে কথা বলব, তারপরেই চলে যাব অপারেটিং সিস্টেম ব্যাপারটা কী — সেটা খায় না মাথায় দেয় — সেই কথায়।

।। দিন এক।।

১।। কাজের কেন্দ্রীয় সংস্থা — সিপিইউ এবং মেমরি

একটা আধুনিক কম্পিউটারের মূল এবং প্রাথমিক অংশগুলো — এক বা একাধিক প্রসেসর, কিছু মূল মেমরি, ডিস্ক, প্রিন্টার, কিবোর্ড, মোডেম বা ল্যান-কার্ড জাতীয় নেটওয়ার্ক সংযোগের উপায়, এবং অন্যান্য ইনপুট-আউটপুটের মাধ্যম। সব মিলিয়ে একটা জটিল ব্যবস্থা। উপাদান হিশেবে আলাদা আলাদা অংশগুলোর আলাদা আলাদা ভূমিকা



এবং চরিত্র নিয়ে কিছু কথা আমরা শূন্য নম্বর দিনেই বলে এসেছি। কিন্তু সেখানে আমাদের মূল বোঝার ব্যাপারটা ছিল — একটা কম্পিউটারের যৌক্তিক গঠন। যুক্তিগত ভাবে কোন ভূমিকা কার। এবারে আমরা বুঝতে চেষ্টা করব বাস্তব ক্রিয়াটা ঘটার জন্যে কম্পিউটারের ভিতরকার আলাদা আলাদা উপাদানগুলো কী ভাবে পরস্পরের সঙ্গে যুক্ত থাকে। ছবিটা দেখুন — 'কাজের উপাদানের নিরিখে কম্পিউটারের গঠন'। স্মার্ট-কার্ড বা হ্যান্ডহেল্ড ডিভাইস ইত্যাদি খুব অন্যরকম মেশিনকে ছেড়ে দিলে অন্য প্রায় সবরকম কম্পিউটারের বেলাতেই এই ছবিতে দেখানো গঠনটা একটা অর্থ বহন করে। ছবিতে কেন্দ্রীয় জায়গায় আছে বাস বা যোগাযোগপথ। এই যোগাযোগপথই আলাদা আলাদা উপাদানগুলোর মধ্যে যোগাযোগ তৈরি রাখে যাতে তথ্যের চলাচল ঘটতে পারে কম্পিউটারের মধ্যে এক অংশ থেকে আর এক অংশে। শূন্য নম্বর দিনেই আমরা দেখলাম, কম্পিউটারের কাজের গোটা রসদটাই হল তথ্য। তথ্য কাজ করে আদেশ মোতাবেক। সেই আদেশগুলোও তথ্য।

কম্পিউটারটা যদি একটা সিস্টেম হয়, তার মধ্যে আবার অনেকগুলো সাবসিস্টেম আছে। সেই সিস্টেমাঙ্গদের ভিতর প্রমুখতম হল — সেন্ট্রাল প্রসেসিং ইউনিট বা সিপিইউ, মেমরি বা স্মৃতি, এবং পেরিফেরাল বা বহিরাঙ্গিক উপাদানগুলো, যেমন ফ্লপি ড্রাইভ, হার্ড ড্রাইভ ইত্যাদি। এই তিন সিস্টেমাঙ্গ মানে সিপিইউ, মেমরি এবং পেরিফেরাল — এরা প্রত্যেকেই অন্যের সঙ্গে সংযুক্ত থাকে বাস দিয়ে। বাস নিয়ে আলোচনা এখন বারবার আসবে। বাস মানে যোগাযোগ ব্যবস্থা। আমরা যেমন যোগাযোগের জন্যে, পথ অতিক্রম করার জন্যে, বাসে চড়ি। এই সাবসিস্টেমগুলো নিজেদের মধ্যে যোগাযোগ করে, তাদের মধ্যে তথ্য যাতায়াত করে, এই বাস দিয়ে। সবাই মিলে পরস্পর যোগাযোগের ভিত্তিতে কাজ করতে পারে এই বাসের মাধ্যমেই। এই ছবিতে দেখুন, দুটো পেরিফেরাল ডিভাইস বা বহিরাঙ্গিক উপাদানকে দেখিয়েছি আমরা — ১ আর ২। ধরুন ১ মানে কিবোর্ড, ২ মানে ফ্লপি, বা যে কোনো অন্য পেরিফেরাল। এই বহিরাঙ্গিক ডিভাইসরা তাদের ডিভাইস কন্ট্রোলারের মাধ্যমে যুক্ত থাকে এই বাস-সংযোগে। ডিভাইস বা উপাদান পিছু একটা করে ডিভাইস কন্ট্রোলার বা নিয়ন্ত্রক। বাসের সঙ্গে যুক্ত থাকে প্রতিটি ডিভাইসের কন্ট্রোলার, তার মারফত সেই ডিভাইসটা। ধরুন, সিপিইউ কোনো তথ্য পাঠাতে চাইছে কোনো একটা ডিভাইসকে, এই অবস্থায় সিপিইউ থেকে তথ্যটা প্রথমে আসবে বাসে, সেখান থেকে ডিভাইসের কন্ট্রোলারে, এবং সেখান থেকে যাবে ডিভাইসে। এই কন্ট্রোলারের কথায় আমরা একটু বাদেই আসছি।

কন্ট্রোলাররা যাদের কন্ট্রোল করে সেই ফিজিকাল ডিভাইস বা ভৌত উপাদানগুলো কাজ করে একদম ভৌত রকমে, র মানে কাঁচা লেভেলে। যেখানে ঘাম ঝরে, রক্ত পড়ে, ঠ্যাং ভেঙে হাড় বেরিয়ে যায়। সেইসব ভৌততা ছেড়ে তাদের কাজকে ভৌতিক মানে সাঙ্কেতিক তথ্যের আকারে নিয়ে আসে এই কন্ট্রোলারনিচয়। পরে ভালো করে বুঝব, ব্যাপারটা অনেকটা এরকম। আপনার মনে একটা আবেগ এল, এক লাইন কবিতা লিখতে চাইলেন, এটা একটা বিমূর্ত ব্যাপার, একটা চিন্তা। আপনি কিবোর্ডে টাইপ করলেন, সেখান থেকে সেটা গেল সরাসরি সিপিইউ-র এক্তিয়ারে, সিপিইউ এবার যে শব্দগুলো অক্ষরগুলো মিলে কবিতা, তাদের তথ্যটা পাঠাল বাসে, বাস থেকে যাবে স্ক্রিন বা কনসোল বা সামনে টাঙানো কাঁচের পর্দার তলে ফুটে ওঠার জন্যে। আপনার কাব্যচিন্তার মত এই তথ্যটাও বিমূর্ত, কিন্তু একটু অন্যরকমে, সিপিইউ তাদের কিছু সংখ্যা বানিয়ে দিয়েছে, যে সংখ্যাগুলোকে পড়ে তাদের পিছনে মূল অক্ষরগুলোকে চিনে নিতে পারবে আপনার ডিভাইসের কন্ট্রোলার। তার জন্যে প্রথমে বাস থেকে তথ্যটা গেল আপনার মেশিনের স্ক্রিন নামক ডিভাইসের কন্ট্রোলারের কাছে, সেই কন্ট্রোলার সেই বিমূর্ত সংখ্যাগুলোকে পড়ল, পড়ে বুঝল সিপিইউ কোন অক্ষরগুলোকে ফুটিয়ে তুলতে চাইছে স্ক্রিনে। কিন্তু কন্ট্রোলার এবার যা পাঠাবে তা আর বিমূর্ত নয়, একদম আছোলা কিছু বৈদ্যুতিক আদেশ, স্ক্রিন জুড়ে থাকা ছোট ছোট চিত্রকোষ বা পিকসেলের মধ্যে এই এই পিকসেলে এত এত মাত্রার বিদ্যুৎ পাঠাও, যাতে স্ক্রিন জুড়ে আপনার লেখা কবিতার লাইনটা ভেসে উঠতে পারে। সেই ভেসে ওঠাটা একদমই ভৌত, আপনার স্ক্রিনের ভৌত গঠনে কোনো গোলযোগ যদি থাকে, সেটা আপনার কবিতার লাইনকে ভেস্তে দেবেই, কবিতাটা যতই উৎকৃষ্ট হোক না কেন।

কন্ট্রোলারগুলোর সঙ্গে যুক্ত এই বাসের সঙ্গেই আবার লাগানো থাকে সিপিইউ এবং মেমরি। সিপিইউ তো লাগানো থাকতেই হবে, সেই কম্পিউটারের সকল কাজের কাজী, এইমাত্র এই কাব্যচর্চার উদাহরণটাকে ভাবুন, আর মেমরি লাগে সিপিইউর প্রতি মুহূর্তের প্রতিটি কাজের জন্যে। তাই এদের দুজনকেও বাসের সঙ্গে সংযুক্ত না-থাকলে কিছুতেই চলবেনা।

সিপিইউ বা সেন্ট্রাল প্রসেসিং ইউনিট হল কম্পিউটারের মূল সক্রিয় অংশ — তথ্য চটকানোর কেন্দ্রীয় সংস্থা। শূন্য নম্বর দিনের ১ নম্বর সেকশনে আমরা কম্পিউটারের পাঁচ রকম উপাদানের ভিতর যাকে শেষ আলোচনা করেছিলাম — সব কাজের নিয়ন্ত্রণ করাই যার কাজ, সেই কেন্দ্রীয় নিয়ন্ত্রণ সংস্থা এই সিপিইউ। এই সিপিইউ-র মধ্যেই ভরা থাকে কম্পিউটারের ঘড়ি। যারা শুধু গোটা গোটা অক্ষরে স্ক্রিনে তারিখ এবং সময় ফুটিয়েই তোলে না, কম্পিউটারের ভিতর চলমান প্রতিটি প্রোগ্রামের সময়কাঠামোকে নিয়ন্ত্রণ করে।

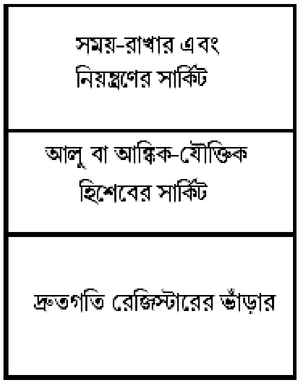
আমরা পরে দেখব প্রচুর প্রক্রিয়া চলে একটা গ্নু-লিনাক্স মেশিনে যারা সরাসরি এই ঘড়িগুলোর দিকে তাকিয়ে থাকে অপলক চোখে — ক্রন, অ্যাট, ইত্যাদি, ঘড়ি ঘড়ি তাদের দিল ধড়কানোই নিয়ম। এই সময়ানুবর্তিতার পাশাপাশি সিপিইউতে থাকে একজন একনিষ্ঠ একাউন্টান্ট। দিবারাত্রি সে শুধু হিশেবই করে, তার নামটা একটু অ্যান্টিক্লাইম্যাক্স যদিও, মানে বাংলায় পড়লে, তার নাম আলু। আমরা আগেই বলেছি। অ্যারিথমেটিক অ্যান্ড লজিকাল ইউনিট। শুধু যোগ বিয়োগ গুণ ভাগই করে না, লজিকাল হিশেবও করে। হ্যাঁ না না, ঠিক না ভুল, ইত্যাদি। আর থাকে নিয়ন্ত্রক সংস্থা। কন্ট্রোল এজেন্সি। প্রোগ্রামের মধ্যেকার নির্দেশগুলো বুঝে যে প্রোগ্রামগুলোকে চালায়। আর থাকে সার্কিট মেমরি থেকে তথ্য নিয়ে আসার। নির্দেশ আর তথ্য — এই দুটোকে ভরা থাকে মূল মেমরিতে মানে র‍্যাম বা রম-এ। প্রয়োজন অনুযায় সিপিইউ তাদের নিয়ে আসে।

বহিরাঙ্গিক উপাদান বা পেরিফেরাল ডিভাইসগুলোর এক একটা করে নিয়ন্ত্রক সংস্থা থাকে। এই কন্ট্রোলারগুলো তাদের নিজের নিজের অধীন উপাদান থেকে, ধরুন ফ্লপির কন্ট্রোলার ফ্লপি থেকে, মাউস কন্ট্রোলার মাউস থেকে তথ্যগুলো নিয়ে আসে বাস-এ। বা তথ্য বার করে নিয়ে যায়। কনসোল কন্ট্রোলার কনসোলে নিয়ে যায় ফুটিয়ে তোলার তথ্য এই বাস থেকেই। ইনপুট আর আউটপুট — তথ্য ঢোকানো বা বার-করার ডিভাইসের এই দু-রকম তথ্যেরই মধ্যস্থতা করে এই কন্ট্রোলাররা। কিবোর্ড বা মাউস জাতীয় ইনপুট ডিভাইস, বা প্রিন্টার, কনসোল জাতীয় আউটপুট ডিভাইস, বা হার্ড ডিস্ক জাতীয় তথ্যের ভাঁড়ার — এই সবগুলো ডিভাইসই সরাসরি কথা বলে তার কন্ট্রোলারের সঙ্গে, কন্ট্রোলার তার পর তথ্যটাকে নিয়ে যায় বাসে। সিপিইউ আর পেরিফেরাল ডিভাইস — এরা কাজ করে একই সঙ্গে — কম্পিউটার আর তার তার ব্যবহারকারীর পারস্পরিকতা বা ইন্টারফেসটাকে বানিয়ে তোলে। ব্যবহারকারী তার মেশিনের সঙ্গে যাতে একটা কথোপকথনে আসতে পারে। তথ্যের সঞ্চারপথটা হয় এরকম — ইনপুট ডিভাইস থেকে কন্ট্রোলার থেকে বাস হয়ে সিপিইউ, বা সিপিইউ থেকে বাস হয়ে কন্ট্রোলার হয়ে আউটপুট ডিভাইস। কখনো আবার সিপিইউ সরাসরি এই তথ্যকে সরাসরি নিজের কাছে না-এনে পাঠিয়ে দেয় মেমরিতে, যাতে পরে সে মেমরি থেকে পড়ে নিতে পারে।

১.১।। সিপিইউ এবং তার রেজিস্টার-কাঠামো

আধুনিক একটা সিপিইউ মানে একটা ছোট্ট সিলিকন চিপ। এই চিপের উপরে এঁকে বা দেগে বা ছেপে দেওয়া লক্ষ লক্ষ ট্র্যানজিস্টর এবং তাদের মধ্যেকার বৈদ্যুতিক সংযোগ — যার গোটাটা মিলিয়ে সিপিইউর মোট সার্কিট। আমরা চিপের ছবি, তার আকার আয়তন নিয়ে কথা বলেছি শূন্য নম্বর দিনেই। চিপের শরীরের ধার ঘেঁষে লাগানো থাকে একশো বা তারও বেশি পিন। এর কিছু পিন সংযোগ করে বাস-এর সঙ্গে, তথ্য যাতায়াত করে এই পিনগুলো বেয়ে।

সিপিইউ-র প্রাথমিক অংশগুলো



আর অন্য পিনগুলো কম্পিউটারের বিদ্যুৎ-লাইন থেকে বিদ্যুৎ বয়ে নিয়ে যায় চিপ অব্দি।

ভৌত গঠনের নিরিখে একটা চিপের শরীর গঠিত হয় অনেকগুলো স্তরে অনেকগুলো জটিল রাসায়নিক এবং বৈদ্যুতিক এবং বৈদ্যুতিন প্রযুক্তিকে মিলিয়ে। কিন্তু, কাজের ছকের ভিত্তিতে একটা সিপিইউ চিপের মূল অংশ সবসময়েই এই তিনটে —

(ক) সময়-রাখার এবং নিয়ন্ত্রণের সার্কিট — সেই সব সার্কিট যারা সময়কে খেয়াল করে, ধারণ করে, এবং করে চলে কন্ট্রোল বা নিয়ন্ত্রণের কাজ, মানে প্রোগ্রাম বা সফটওয়ারগুলোর মধ্যে লিপিবদ্ধ নির্দেশগুলোকে লাগু করে। এই টাইমিং এবং কন্ট্রোল সার্কিট হল কম্পিউটারের জরুরিতম অঙ্গ সিপিইউ-র আবার সবচেয়ে জরুরি অংশ। ধরুন শেষ অব্দি কম্পিউটারের কম্পিউটার অংশ কতটুকু? তার উত্তর হবে এই টাইমিং এবং কন্ট্রোল সার্কিট। দিন শূন্য থেকে ক্যালকুলেটর আর কম্পিউটারের পার্থক্যের আলোচনা মনে করুন। একটা সময়-রাখা এবং নিয়ন্ত্রণের সার্কিটের কাজের গড়ন অনেকটা এইরকম হতে পারে —

repeat — এর মানে, এইখান থেকে এর পরের অংশ বারবার করে চলবে সিপিইউ, হল্ট অব্দি।

fetch next instruction from memory — মেমরিতে তুলে রাখা প্রোগ্রামের পরের নির্দেশটা পড়ো।

decode instruction — নির্দেশ অনুযায়ী ঠিক কোন তথ্যকে কোন সার্কিট দিয়ে চটকাতে হবে সেটা স্থির করো।

fetch from memory additional data — মেমরি থেকে, যদি লাগে, আরো তথ্য নিয়ে এসো।

execute the instruction — সঠিক সার্কিটে সঠিক তথ্যটা পৌঁছে দাও, যাতে সে কাজ করতে পারে।

until halt — এর মানে এই অব্দি এসে বৃত্তটা সম্পূর্ণ হল, রিপিট থেকে যা শুরু — এটা একটা লুপ, পরে আসব আমরা।

শুধু এই ‘ফেচ-ডিকোড-একজিকিউট’ বা ‘আনো-বোঝো-পালন-করো’ জাতীয় নির্দেশ-বৃত্ত বা লুপই নয়, এই টাইমিং এবং কন্ট্রোল সার্কিটেরই আভ্যন্তরীন হল সেইসব সার্কিট যারা নির্দেশগুলোকে ডিকোড করে মানে বোঝে, আর সেই নির্দেশের মধ্যে আবদ্ধ থাকা ঠিকানাগুলোকে চিনতে পারে। নইলে দেখুন না, সে যদি নিতান্তই সময় রাখা আর নিয়ন্ত্রণের বাইরে আর কিছু না-পারে, তাহলে ওই বৃত্তটার তিন-নম্বর লাইনে, মানে ‘ডিকোড’-এ এসে তো আটকে যাবে — ভাই বোঝার দায়িত্ব তো আমার নয় — আমার দায়িত্ব তো করা? আর বোঝার মধ্যেই রয়ে যাচ্ছে কিছু ঠিকানা চিনে নেওয়া, কারণ, এর পরের লাইনেই আসছে, প্রয়োজন মোতাবেক আরো তথ্য নিয়ে এসো। নিয়ে এসো তো বললাম, আনবে কোত্থেকে? আর আধুনিক একটা চিপ, সাইজ তো আগেই দেখলেন, ভারি চিকণ জিনিষ, হনুমান নয় যে গন্ধমাদন উপড়ে আনবে বিশল্যকরণীর সঠিক বাস-রুট পায়নি বলে। এই যে লুপ নিয়ে একটু কথা আমরা বলে নিলাম, এই প্রসঙ্গটা বারবার ফিরে ফিরে আসবে আমাদের গোটা পাঠমালাটা জুড়ে।

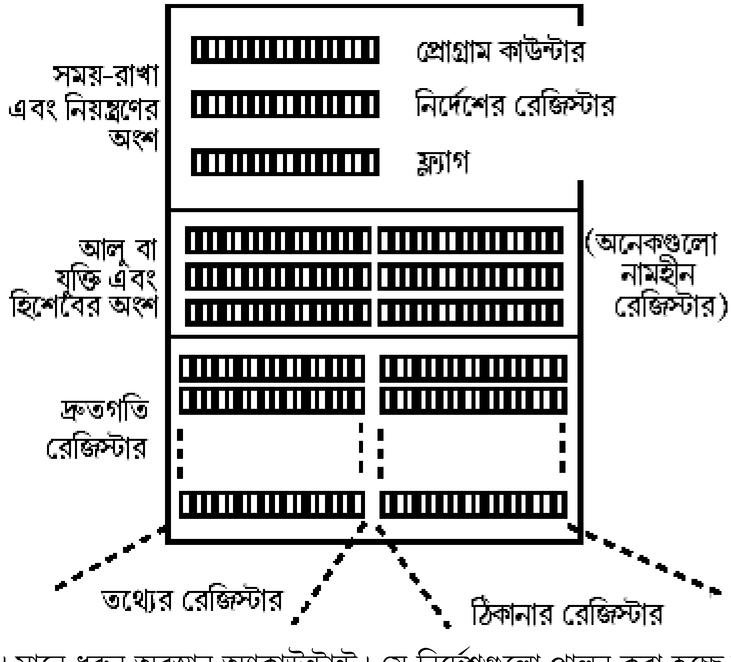
(খ) আলু বা আঙ্কিক-যৌক্তিক হিশেবের সার্কিট — আলু বা অ্যারিথমেটিক লজিক ইউনিটের মধ্যে থাকে সেইসব সার্কিট যারা আঙ্কিক তথ্যকে ইধার-কা-মাল-উধার, মানে ম্যানিপুলেট করে, নাড়াচাড়া করে। যোগ বা গুণ করার জন্যে নির্দিষ্ট সার্কিট আছে। অনেকসময়েই এই সার্কিটগুলো শুধু এক রকমের না, একাধিক রকমের হয়। কিছু সার্কিট হয় যারা পূর্ণসংখ্যা মানে ইন্টিজারদের নিয়ে নাড়াচাড়া করে। কিছু সার্কিট করে রিয়াল নাম্বার বা বাস্তব সংখ্যাকে। কিছু সার্কিট আছে তুলনা করার জন্যে। একটা নির্দিষ্ট ভ্যারিয়েবল একটা নির্দিষ্ট মানের চেয়ে ছোট না বড় না সমান — প্রোগ্রামগুলোর এটা বোঝার দরকার পড়ে হরহামেশাই। সেই কাজটা করে দেন এই আলুবাবু। ডেটা বা তথ্যের একটা নির্দিষ্ট আকৃতি বা বিট প্যাটার্ন আছে কিনা সেটারও বিচার করেন। দরকার পড়লে সেই আকৃতি বা বিট-প্যাটার্নটা বদলাতেও পারে আলুবাবুর লজিক সার্কিটরা।

(গ) দ্রুতগতি রেজিস্টারের ভাঁড়ার — কম্পিউটার তার প্রয়োজনের বেশিরভাগ তথ্যটাই রাখে মেমরিতে। কিন্তু সামান্য কিছু তথ্য রাখার ব্যবস্থা থাকে সিপিইউ-র ভিতরেই — রেজিস্টারে। মূল মেমরিটা যেখানে কোটি কোটি তথ্য একক ধরে রাখতে পারে, সেখানে এই রেজিস্টার পারে ধরুন ষোলটা তথ্য একক। সাত নম্বর দিনে, ভারচুয়াল মেমরির প্রসঙ্গে আমরা এই বিভিন্ন ধরনের মেমরির একটা তুলনামূলক আলোচনা করব। একটা সিপিইউ রেজিস্টার রাখতে পারে কম্পিউটারের একটা ওয়ার্ড বা শব্দর যা মাপ ঠিক সেই পরিমাণ তথ্য। ওয়ার্ড এর সংজ্ঞা দেখুন শূন্য নম্বর দিনের শেষে তথ্য পরিমাপের এককে। আর একবার মনে

> করিয়ে দিই, একটা বিট মানে নিছক একটা ০ বা ১, একটা বাইট মানে আটটা বিট-এর একটা সমাহার, একটা ওয়ার্ড কম্পিউটার থেকে কম্পিউটারে বদলায়, ষোল বিট হতে পারে, মানে দুই-বাইট বা বত্রিশ বিট হতে পারে, মানে চার বাইট, বা এমনকি তার বেশিও হতে পারে। বেশিরভাগ আধুনিক সিপিইউ-র বেলায় এই রেজিস্টারের মাপ হল বত্রিশ বিট করে। আলুর সার্কিট এমনভাবেই তৈরি এবং সমাবদ্ধ যে আলুর কাছে যে তথ্য যায়, বা আলু থেকে যে তথ্য বেরোয়, তার অনেকটা বা পুরোটাই — গোটা ইনপুট আউটপুটটাই ঘটে সিপিইউ রেজিস্টার মারফত। তথ্যটা মেমরি থেকে তুলে এনে চকিত চোখের চঞ্চল চাহনির মত একটা ছোট্ট মুহূর্ত ধরা থাকে সিপিইউর রেজিস্টারে, তারপরেই মালটা চলে যায় আলুর দখলে। আর হিশেবের কাজ শেষ হয়ে গেলে ফেরত এনে তুলে রাখা হয় মূল মেমরির আলমারিতে। সিপিইউর কিছু রেজিস্টার থাকে এই তথ্য মানে বিট-মানগুলোকে ধরে রাখার জন্যে, আর কিছু রেজিস্টার থাকে সিপিইউ কাজ করে চলাকালীন মেমরির কোন কোন ঠিকানায় কী কী করতে হবে সেই হিশেব নিকেশ করার জন্যে। রেজিস্টারগুলো কী ভাবে ব্যবহার হবে তার ভিত্তিতে সিপিইউগুলোর গঠনের তফাত হয়, কিন্তু, মোটামুটিভাবে, বেশিরভাগ সিপিইউতেই আটটা বা তার বেশি রেজিস্টার থাকে তথ্য রাখার, আর আরো আটটা থাকে ওই ঠিকানার জন্যে।

এই তথ্য আর ঠিকানার রেজিস্টার ছাড়াও আরো রেজিস্টার থাকতে পারে একটা সিপিইউতে — আলু আঙ্কিক হিশেব করে চলাকালীন ছোট ছোট 'হাতে-রইল-পেন্সিল'-এর জন্যে। আবার সময়-রাখার এবং নিয়ন্ত্রণের সার্কিটেও থাকে কিছু রেজিস্টার — নিয়ন্ত্রণ সংক্রান্ত তথ্য রাখার জন্যে। সঙ্গের ছবিতে দেখুন টাইমিং এবং কন্ট্রোল ইউনিট বা সময়-রাখা এবং নিয়ন্ত্রণের অংশে তিন রকম রেজিস্টার হয় — প্রোগ্রাম কাউন্টার বা প্রোগ্রামের হিশেবরক্ষক, ইন্সট্রাকশন বা নির্দেশের রেজিস্টার এবং ফ্ল্যাগ। এই প্রোগ্রাম কাউন্টার ধরে রাখে মেমরির সেইসব এলাকার ঠিকানা যাতে ঠিক এরপরেই যে নির্দেশটা পালন করতে হবে সেটা যেখানে লেখা আছে। নির্দেশ বা ইন্সট্রাকশন রেজিস্টার ধরে রাখে এই মুহূর্তে যে নির্দেশটা পালন করা হচ্ছে তার বিট সমাহার। এছাড়াও বেশিরভাগ সিপিইউতেই থাকে

সিপিইউ রেজিস্টারগুলোর আলাদা আলাদা কাজ



একটা ফ্ল্যাগ রেজিস্টার। মানে ধরুন অবস্থান অ্যাকাউন্টান্ট। যে নির্দেশগুলো পালন করা হচ্ছে তাদের অবস্থাটা লিখে রাখে এই ফ্ল্যাগ। একটা বিট থাকে যেটা দেখায় যে ওই পালনমান নির্দেশটা সিপিইউতে এসেছে কোনো সাধারণ প্রোগ্রাম থেকে না সরাসরি অপারেটিং সিস্টেম থেকে।

এই অস (OS — **O**perating-**S**ystem) বা অপারেটিং সিস্টেম হল কম্পিউটারের ভিতরে সকল মাস্তানের বাবা মাস্তান, অন্য সকল প্রোগ্রামাস্তানেরা যেখান থেকে প্রবাহিত হয়, সেই নেতার নেতা, মানে ধরুন, রাজনীতির মুখ্যমন্ত্রী বলা যায়। তার অনেক কিছু করার রাইট থাকে যা অন্য প্রোগ্রামের কাছে রং, ওখানে কোনো রংবাজি চলবে না কাকা।

আমাদের পরে এটা খুব ভালো করে বুঝতে হবে, বিশেষত একটা গ্নু-লিনাক্স সিস্টেমের সাপেক্ষে। এখানে যে কথাগুলো আমরা বলছি, সেগুলো সবটাই হল ভিতখোঁড়া, পরে এর উপরে আমরা গ্নু-লিনাক্স অপারেটিং সিস্টেমের কাঠামোটা গজিয়ে উঠতে দেখব। আর খুব বেশি করে এই সিপিইউ রেজিস্টারের আর ফ্ল্যাগের ধারণাগুলো যেখানে জরুরি হয়ে ওঠে, সেটা হল মূলত গ্নু-লিনাক্স অ্যাজ এ প্রোগ্রামিং এনভায়রনমেন্ট — যা তথাগতর ভবিষ্যদ্বানী অনুযায়ী ৩৫ বা ৩৬ নম্বর দিনের আগে আসার চান্স নেই। তার মানেই চান্স নেই। কারণ এই বইটা তার চেয়ে অনেক ছোট হবে, প্রোগ্রামিং অর্থে একটু ব্যাশ স্ক্রিপ্টিং আসবে, তাও এই পাঠমালার দশ নম্বর মানে শেষ দিনে গিয়ে।

ফ্লাগ ব্যাপারটাও অসম্ভব জরুরি হয়ে ওঠে প্রোগ্রামিং বোঝার বেলাতেই। ইন ফ্যাক্ট এই সেকশনের ছবির ছকগুলো এবং বক্তব্যেরও বেশিরভাগটাই নেওয়া একটা সিপ্লাসপ্লাস প্রোগ্রামিং-এর বই থেকে, যার নামটা আমার মনে নেই, ডাউনলোড করা ফাইলটাও ভুল বশত উড়ে গেছে। নেটে সিপ্লাসপ্লাস দিয়ে সার্চ মারলে পেয়ে যাওয়ার কথা, নয়ত কারোর লাগলে আমার মূল নোটসটা দিতে পারি। সেখান থেকে নেওয়া নোটসটাই এখানে কাজে লাগছে। না ভাই, ঝাড়া বলবেন না, কানাকে কানা বলতে নেই, আগেই বলেছি, অন্নু মালিক আমাদের রোল মডেল। আপনাদের কাছে খিস্তি খাওয়ার ভয়ে দেখুন, প্রতিটি দিনের শেষে লেখা 'রচনা ও সংকলন', শুধু 'রচনা' নয় বস।

যাই হোক, যে ফ্লাগ নামক রেজিস্টারগুলোর কথা বলছিলাম, এদের উপর পুরো স্বৈরতান্ত্রিক নিয়ন্ত্রণ চলে অপারেটিং সিস্টেমের। এই অপারেটিং সিস্টেমের অনেক বাড়তি অধিকার থাকে, শুধু এই ফ্লাগদের উপর নয়, বহিরাঙ্গিক বা পেরিফেরাল উপাদানগুলোর কন্ট্রোলারদের কাজকর্মের উপরও, যা অন্য সব আমপ্রোগ্রামের থাকেনা। সাধারণ প্রয়োগমূলক বা অ্যাপ্লিকেশন প্রোগ্রামগুলোর যখন এদের কোনোটাকে ব্যবহার করার প্রয়োজন পড়ে, তারা আবেদন জানায় কারনেলের বা অপারেটিং সিস্টেমের কাছে। কারনেল হল সকল কাজের কাণ্ডারী, সকল কোষের ভাণ্ডারী। সে ঠিক করে দেয় কখন কোন প্রোগ্রাম কতটুকু অব্দি এদের ব্যবহার করতে পারবে। পরের দিন, মানে, এই পাঠমালার দুই নম্বর দিনে এটা নিয়ে আর একটু আলোচনা করব আমরা, তারপর কম্পিউটার বিবর্তনের ইতিহাসের সঙ্গে মিলিয়ে তাদের বুঝব চার নম্বর দিনে গিয়ে। কারনেলের এই চূড়ান্ত নিয়ন্ত্রণটাই ঘোষিত থাকে ফ্লাগ রেজিস্টারদের একটা বিটে। অন্য বিটগুলোর গায়ে লেখা হয় আলুবাবুর বিভিন্ন তুলনামূলক কাজকর্মের ফসলাবলী। ধরুন একটা ভ্যারিয়েবল যদি একটা সংখ্যার সমান হয় — সেটা লেখা থাকবে একটা বিটে, অন্য একটায় লেখা থাকবে যদি সেটা বড় হয় — ইত্যাদি। মানে একটা বিট হল নিয়ন্ত্রণের বিট, অন্যগুলো হল কাজের ফলাফলের বিট।

আসলে একটা অন থাকা কম্পিউটার মানে তো কিছু চলমান প্রোগ্রামের সমষ্টি। এবং কিছু ভৌত উপাদানের শরীরে প্রতিমুহূর্তে লেখা হতে থাকা আঁকা হতে থাকা এই প্রোগ্রামগুলোর সঞ্চারপথ। আর একটা প্রোগ্রাম মানেই কিছু পরপর লিখে রাখা নির্দেশের তালিকা। যা মেমরিতে তুলে ফেলা হয়েছে এবং সিপিইউ পরপর তাকে পালন করে যাচ্ছে। আর এই করতে গিয়ে যে যে পেরিফেরাল ডিভাইসকে তার যে যে ভাবে যতক্ষণ ধরে কাজে লাগানো দরকার সেই সেই ভাবে তাদের সক্রিয় করে তোলা হচ্ছে। ঠিক এই কাজগুলোকে পরপর চালিয়ে যাওয়াই সিপিইউর রেজিস্টারগুলোর কাজ। আপাতত এটা যথেষ্ট হয়েছে — এবার দু-একটা কথা বলে নেওয়া যাক সিপিইউর সঙ্গে মেমরির সম্পর্ক নিয়ে। আমরা এতবার করে বলছি, মেমরির ঠিকানা, মেমরি থেকে তথ্য তোলা — এর ঠিক মানেটা কী। সিপিইউ ঠিক কী ভাবে এটা করে।

১.২।। মেমরি এবং তথ্য

আমরা আগেই বলেছি, কম্পিউটারের দু-রকম মেমরি — রম (ROM — **R**ead-**O**nly-**M**emory) আর র‍্যাম (RAM — **R**andom-**A**ccess-**M**emory)। রম মানে যেখান থেকে সিপিইউ শুধু পড়ে। আর র‍্যাম মানে যেখানে, যেমন দরকার, পড়ে বা লেখে। মেশিনের প্রাথমিকতম রম আর র‍্যাম থাকে মাদারবোর্ডের চিপে।

আর এর বাইরে আর এক রকম মেমরি হল তথ্যের ভাঁড়ার বা স্টোরেজ, হার্ড ডিস্ক, সিডি, ফ্লপি ডিস্ক, জিপ ড্রাইভ জাতীয়। যাকে সচরাচর আমরা যখন কম্পিউটারের মেমরি বলে উল্লেখ করি তার মধ্যে ধরিনা। র‍্যাম-এর নাম র‍্যাম হয়েছে মূলত এই ধরনের ভাঁড়ার বা স্টোরেজ ডিভাইস থেকে একে আলাদা করতে গিয়ে। এখান থেকে যে কোনো ভাবে তথ্য পড়া যায় — যে কোনো বিটের পরে যে কোনো বিট — এই অর্থে এটা র‍্যানডমলি অ্যাকসেসিবল।

রম সচরাচর একটা কম্পিউটারের অপারেটিং সিস্টেমের প্রাথমিকতম অংশগুলোকে ধরে রাখে। অপারেটিং সিস্টেম নিয়ে একেবারে কিছু না-বলে এদের নিয়ে কথা বলা শক্ত। বারবার খুব অসুবিধা হচ্ছে কথা বলতে গিয়ে, কিন্তু কিছু করার নেই, ঠিক এরের পরের ২ নম্বর সেকশন থেকেই আমরা আসব অপারেটিং সিস্টেমের আলোচনায়। দুই থেকে পাঁচ নম্বর দিনের গোড়া অব্দি চলবে এই অপারেটিং সিস্টেম কী করে গজিয়ে উঠল তার ইতিহাস। সেই ইতিহাস যৌথ ভাবে কম্পিউটারের হার্ডওয়ার এবং সফটওয়ারের ইতিহাস। তারপর গোটা পাঠমালা জুড়ে বারবার আসতেই থাকব একটা বিশেষ রকম অপারেটিং সিস্টেম — গ্নু-লিনাক্স অপারেটিং সিস্টেমের আলোচনায়।

অপারেটিং সিস্টেমটাও একটা প্রোগ্রাম। সব প্রোগ্রামের কেন্দ্রীয় প্রোগ্রাম। এই অপারেটিং সিস্টেমের প্রায় গোটাটাই লেখা থাকে হার্ড ডিস্কে। কম্পিউটার অন করার পর থেকে কাজ করার অবস্থায় আসা অব্দি গোটা প্রক্রিয়াটার নাম বুট। এই বুট করাকালীন কী ভাবে রম চিপে লেখা থাকা নির্দেশ মেনে হার্ড-ডিস্কে লিখে রাখা অপারেটিং সিস্টেমকে সক্রিয় এবং জীবন্ত করে তোলে সিপিইউ, প্রাণ-প্রতিষ্ঠা করে সিলিকন-তার-চাকতি-প্লাস্টিক-স্ক্রু ইত্যাদি দিয়ে তৈরি ওই দামী এবং জটিল খেলনায় — সেই আলোচনা করব আমরা পাঁচ নম্বর দিনে।

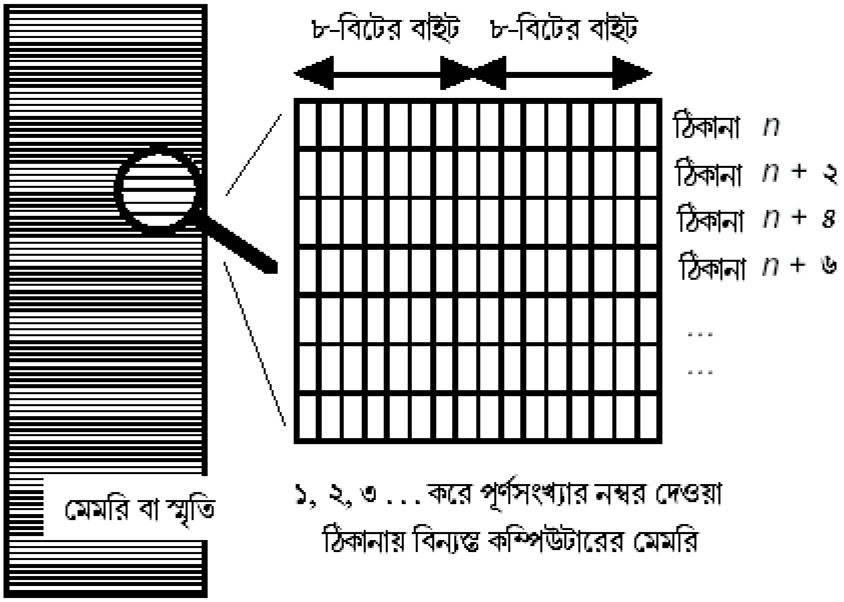
শূন্য নম্বর দিনে আমরা উল্লেখ করেছি, একটা কম্পিউটারে স্বভাবত র‍্যামের মোট পরিমাণের তুলনায় রমের পরিমাণ যৎসামান্য। মেমরির সিংহতমভাগই এই র‍্যাম। র‍্যাম তার চিপের পর চিপ জুড়ে ধরে রাখে অপারেটিং সিস্টেমের মোট কোড এবং তথ্য। কোড বলতে এখন থেকে আমরা বুঝব প্রোগ্রামকে, প্রোগ্রাম মানে তো পরপর লিপিবদ্ধ নির্দেশ — সেই নির্দেশগুলোর ক্রমানুসারী তালিকাটাকে এখন থেকে আমরা কোড বলে ডাকব। সেটাই চালু নাম। শুধু এই অপারেটিং সিস্টেমের কোড এবং সেই কোডগুলো চালানোর জন্যে প্রয়োজনীয় তথ্যই নয়, অপারেটিং সিস্টেম এরপর যে প্রোগ্রামগুলোকে চালাবে সেই প্রোগ্রামগুলোর কোড এবং তথ্যও একই ভাবে তুলে নিতে হয় র‍্যামকে। মেমরির পরিমাণের একক, আমরা বলেছি, বিট বাইট এবং ওয়ার্ড। বিট বা বাইট তো সব কম্পিউটারেই এক, কিন্তু ওয়ার্ডের মাপ কম্পিউটার থেকে কম্পিউটারে বদলায়। দুই বাইট থেকে চার বা আট বাইট, ১৬ বা ৩২ বা ৬৪ বিট, বা হয়ত আরো বেশি আসবে আগামী দিনে। মেমরির সাইজ উল্লেখ করা হয় বাইট দিয়ে। বা ১০২৪ গুণ ১০২৪ মানে ১০৪৮৫৭৬ বাইট বা এক মেগাবাইট দিয়ে। বিট দিয়ে উল্লেখ করা মানে ধরুন ১২৮ এমবি-র একটা র‍্যামের সাইজ দাঁড়াবে ১০৭৩৭৪১৮২৪ বিট — দেখে নিয়ে আর না-তাকিয়ে একবারে বলুন তো আপনার মেশিনের র‍্যামের সাইজ কত বিট?

এই র‍্যাম বলুন, রম বলুন, স্টোরেজ ডিভাইস বলুন, সবাই শেষ অব্দি কাজ করে বিট দিয়ে। একটা বিট মানে একটা ছোট্ট খোপ — যাতে থাকতে পারে হয় ০ নয় ১। তার মানে খোপটার দুটো ভৌত অবস্থা থাকতে পারে, একটা ভৌত অবস্থা হাজির করবে শূন্য, অন্য ভৌত অবস্থা হাজির করবে এক। কী ভাবে এই অবস্থাটা তৈরি হবে তার অনেকরকম প্রকৌশল আছে। চার নম্বর দিনে আমরা যখন প্রাচীনতম কম্পিউটারগুলোর আলোচনা করব তাতে আসবে ভালভসেট দিয়ে বানানো মেশিনের কথা, যেখানে একটা পারদের টিউবের ভিতর একটা বিদ্যুৎ-উপস্থিতি মানে এক, আর তাই, না-থাকা মানে শূন্য। ভাঁড়ার তখন রাখা হত মূলত মেশিনের বাইরে, কাগজে বা কার্ডে, এই কার্ডে আবার একটা ফুটো থাকা মানে শূন্য, ফুটো না-থাকা মানে এক। ক্যাসেট টেপের ফিতেয় রাখা মানেও তাই — কোনো একটা চৌম্বক উপস্থিতি মানে ১, অনুপস্থিতি মানে ০। এর পরে এসেছিল চৌম্বক অক্সাইডের ছোট ছোট লুপ — যার নাম ছিল কোর, যার উত্তর-দক্ষিণ মেরু-সংস্থান দিয়ে হাজির করা হত ০/১। এই নামটা আজো চলছে। ইউনিক্স জগতে, ওপন-সোর্স-কোড ইউনিক্স মানে লিনাক্স জগতেও, এখনো র‍্যামকে অনেকসময়ই ডাকা হয় কোর বা কোর মেমরি বলে। যদিও এখন এই প্রকৌশল আমূল বদলে গেছে। তবে হার্ড ডিস্কে যেভাবে তথ্য রাখা হয় সেটাও এক ধরনের একটা চৌম্বক পদ্ধতি।

প্রকৌশলের সবচেয়ে নাটকীয় বদলটা এসেছিল ইন্টিগ্রেটেড সার্কিট থেকে। শূন্য নম্বর দিনে আমরা ছবি দিয়ে দেখিয়েছি — এখনকার র‍্যাম তৈরি হয় এই আইসি বা ইন্টিগ্রেডেট সার্কিট চিপ দিয়ে। এর ভিতর খুব প্রাথমিক একটা সার্কিটকে বলে ফ্লিপ-ফ্লপ সার্কিট। তার দিয়ে জোড়া চারটে ট্রানজিস্টর মিলিয়ে সার্কিটটা — একটা একক, এটা অন থাকা মানে ১, আর অফ থাকা মানে ০। অর্থাৎ এই এককটা একটা এক বিট তথ্যকে ধরে রাখতে পারে। আর একটু বড় সাইজের সার্কিট, আটটা ফ্লিপ-ফ্লপের, রাখতে পারে এক বাইট তথ্য। এরকম অজস্র সার্কিট মিলিয়ে

মিলিয়ে তৈরি হয় র‍্যাম চিপ যা লক্ষ লক্ষ বাইট তথ্যকে ধরে রাখতে পারে। এরকম বেশ কয়েকটা চিপ মিলিয়ে তৈরি হয় একটা কম্পিউটারের র‍্যাম। শূন্য নম্বর দিনে র‍্যামের ছবিটা মনে করুন।

এবারে এই র‍্যামের কাজ করার ছকের ছবিতে দেখুন — একটা র‍্যামের আভ্যন্তরীন কাঠামো। যেখানে ওয়ার্ড ধরেছি আমরা ষোল বিটের মানে দুই বাইটের। এবার ছবির সঙ্গে মিলিয়ে ভাবুন, একটা র‍্যামকে ভাবা যেতে পারে পরপর ওয়ার্ডের পর ওয়ার্ডের একটা একটা একমাত্রিক সজ্জা। সজ্জাটার সঙ্গে মিলিয়ে প্রত্যেকটা ওয়ার্ডের একটা অবস্থান আছে — মানে একটা সূচক সংখ্যা বা ইনডেক্স। এক দুই তিন করে।



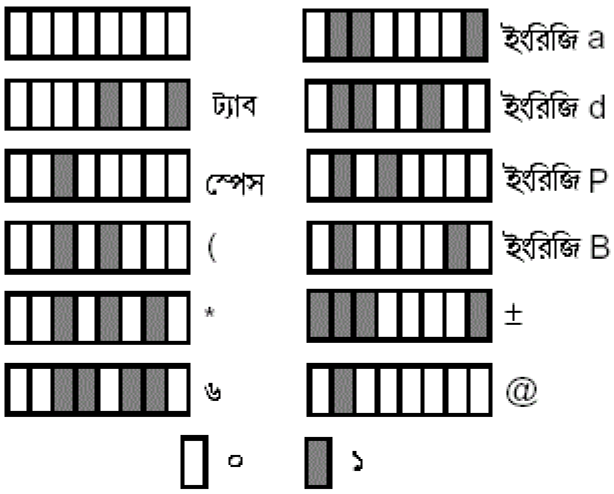
এই সূচক বা ইন্ডেক্সটাই হল র‍্যামের মধ্যের প্রতিটি ওয়ার্ডের আলাদা আলাদা ঠিকানা। এবং একটা ওয়ার্ডের মধ্যে আছে দুটো করে বাইট। বেশিরভাগ কম্পিউটারেই ব্যবস্থা থেকে আলাদা ভাবে একটা একক বাইটকেও পড়ে ফেলার। তার মানে প্রতিটি ওয়ার্ড বাড়ছে দুই বাইট করে। ২, ৪, ৬ — এই ভাবে। যদি সেই কম্পিউটারে ওয়ার্ড হয় চার বাইটের তখন এটা এগোবে ৪, ৮, ১৬ — এই ভাবে।

এই র‍্যামের পরে আসে ক্যাশে-র (Cache) প্রসঙ্গ। ক্যাশের উল্লেখ আমরা শূন্য নম্বর দিনেই করেছিলাম। কম্পিউটারের পাঁচ রকম স্মৃতির কথা বলতে গিয়ে। ক্যাশে শব্দটার ব্যুৎপত্তিগত অর্থ হল গোপন ভাণ্ডার। ক্যাশে মেমরিও ঠিক সেইরকমই। সাধারণ প্রোগ্রামারকে এর হদিশ রাখতে হয়না, আর প্রোগ্রাম না-বানিয়ে সে শুধু যদি প্রোগ্রাম ব্যবহারকারী হয় তাহলে তো জানার সম্ভাবনা আরো কম। ক্যাশে মেমরি থাকে কম্পিউটারের হার্ডওয়ারে, রাখা হয় শুধু দ্রুতগতিতে সিপিইউকে কাজ করানোর তাগিদে। পরে আমাদের বারবার আসতে হবে এগুলোর কথায়। তখন আমরা আরো জানব। ভারচুয়াল মেমরি বা ভৌতিক স্মৃতি — যা একটা গ্নু-লিনাক্স সিস্টেমের সবচেয়ে জোরের জায়গাগুলোর একটা, সেই ভারচুয়াল মেমরি, এবং তার ভৌত আকার মানে হার্ডডিস্কে লেখা সোয়াপ ফাইল বোঝার সময় আট নম্বর দিনে সবচেয়ে বিশদভাবে আসব আমরা এই আলোচনায়।

শুধু এবার মেমরি নিয়ে আমাদের এই আলোচনা-টাকে মিলিয়ে নিন শূন্য নম্বর দিনের একদম শেষে তথ্য তৈরির উপায় হিশেবে অ্যাস্কি কোডের আলোচনার সঙ্গে। সঙ্গের ছবির নিরিখে। আমরা আগেই বলেছি, কম্পিউটার তার ব্যবহার্য সমস্ত তথ্যকেই নাড়াচাড়া করে এবং চেনে বিট-সমাহার বা বিট-প্যাটার্ন দিয়ে। একটা বাইটে থাকে আটটা বিট। আটটা বিট মানে, আমরা দেখিয়েছি ২৫৬-টা আলাদা আলাদা বিট-সমাহার। নিজেই নিজেকে পরখ করে নিন, আপনি মনে করতে পারছেন কি, কোথা থেকে গজালো এই ২৫৬ সংখ্যাটা? জিএলটি ইশকুলে যাদের উপর অত্যাচার করার সুযোগ আমার হয়েছে, তাদের সবাইকে এই ধরনের প্রশ্ন মুহুর্মুহু এই ধরনের প্রশ্ন করে চমকে দিতে খুব ভালোবাসি আমি। এখানে আপনি নিজেই নিজেকে করুন। যা আলোচনা হয়েছে এখনো তাতে এটা আপনার মনে করতে পারা উচিত। একটা ইঙ্গিত দিই — ‘২’ সংখ্যাটাকে ‘৮’ সূচকে তুলে, মানে ‘$২^{৮}$’ হল ২৫৬। এবার কি বুঝতে

পারছেন? একবার হিশেব করে দেখুন তো, বিটের সংখ্যা এক এক করে বাড়িয়ে, কটা করে আলাদা আলাদা সমাহার মানে সংখ্যা রাখা যাচ্ছে। আর এই সংখ্যাগুলো দিয়েই তো আমাদের মেশিনকে বোঝাতে হয় আর যা যা বোঝাতে চাই তার সমস্ত কিছু।

এই ২৫৬-টা সম্ভাব্য বিট-সমাহারের কয়েকটা আমরা এখানে দেখালাম। এক বাইট মেমরি-ভূমিতে যাদের রপ্তানি করা যায় সেরকম কিছু চিহ্ন এবং অক্ষর।



সম্ভাব্য ২৫৬-টা বিট-সমাহারের কয়েকটা

অপারেটিং সিস্টেমের আলোচনা শুরুর আগে সিপিইউ এবং মেমরি কিছু কথা আমরা বলে নিতে চেয়েছিলাম। সেই এক নম্বর দিনের এক নম্বর সেকশন এখানে শেষ। এবার আমরা যাব অপারেটিং সিস্টেম কাকে বলে তা নিয়ে একদম গোড়ার কিছু কথায়। যেখান থেকে আমরা অপারেটিং সিস্টেমের আলোচনা শুরু করতে পারব — যা এখন বেশ কয়েকদিন ধরে চলবে। অপারেটিং সিস্টেমের ইতিহাসের আলোচনা ধরে আমরা ইউনিক্স-এ পৌঁছব, তারপর ইউনিক্স থেকে মিনিক্স থেকে ওপন-সোর্স ইউনিক্স মানে গ্নু-লিনাক্স। যেখানে আমরা শেষ অব্দি পৌঁছতে চাই। আমাদের এই জিএলটি ইশকুলের পাঠমালায়। এই অব্দি আমরা হার্ডওয়ার নিয়ে কম্পিউটার সিস্টেম নিয়ে এমন বহু কথা বলেছি এবং আরো বলব যেগুলো সচরাচর কম্পিউটার ব্যবহারকারী হিশেবে আমাদের না-জানলেও চলে।

কিন্তু, প্রথমত আমরা অনেকেই যেখানে কাজ করে অভ্যস্ত সেই উইনডোজ পরিমণ্ডলের সঙ্গে গ্নু-লিনাক্স পরিমণ্ডলের একটা বড় পার্থক্য এই যে এখানে, গ্নু-লিনাক্সে যদি কেউ সিস্টেমকে তার গভীর অব্দি বুঝতে এবং ব্যবহার করতে চায় তার সেই সুযোগ আছে — যা উইনডোজ-এ নেই, গ্নু-লিনাক্স একটা কমিউনিটি, একটা আন্দোলন, উইনডোজ তা নয়। মাইক্রোসফট নামক একটা বহুজাতিক প্রতিষ্ঠানের ব্যবসার নাম উইনডোজ। তাই, তার ব্যবসার স্বার্থেই সে আপনাকে জানতে দেবে না, তার প্রোগ্রামগুলো একদম ভৌত স্তরে ঠিক কী ভাবে কাজ করে। পরে চার এবং পাঁচে এটা আমরা খুঁটিয়ে বুঝব। আর সেই জানাটা জানতে দেবেনা বলেই অপারেটিং সিস্টেমকে খুব তলিয়ে বোঝার চেষ্টা করাটাই উইনডোজ প্রতিবেশে একটা বোকামি, একটা জায়গায় গিয়ে সেটা আটকে যেতে বাধ্য। কিন্তু গ্নু-লিনাক্সে তা নয়, সবটাই খোলা প্রকাশিত মুক্ত, যতদূর আপনি জানতে চান, ততদূরই জানতে পাবেন। তাই যে জানাটা উইনডোজ প্রতিবেশে সম্পূর্ণ অকেজো সেটারই এখানে প্রচুর ব্যবহার আছে।

দুই, ঠিক ভাবে গ্নু-লিনাক্স বোঝাটাও উইনডোজ-এর চেয়ে অনেক বেশি রকম মেশিনমন্যতা দাবি করে। এর চেয়ে অন্য ভাবেও, মানেও ঠিক উইনডোজ-এর মত করেই, উইনডোজের চেয়ে আরো অনেকটা উন্নত এবং দক্ষ এবং নিরাপদ অন্যরকম একটা উইনডোজ হিশেবেই, গ্নু-লিনাক্সকেও, ব্যবহার করাই যায়, তাতে আর অতো মেশিন বোঝার দরকার পড়েনা। কিন্তু সে তো অনেকটা এমনি এমনি খাওয়া, নাতি বয়স থেকে দাদু বয়স অব্দি আপনি এমনি এমনিই খান, তার জন্যে এত লেখাপড়ার ছাই দরকার কিসে? এই বুড়ো বয়সে গ্নু-লিনাক্স শিখতে এসে আমার মনে হয়েছিল — সত্যিই — ঘরের কাছে আরশিনগরে এই যাদুকর পড়শির কথা — আর একটু আগে জানিনি কেন?

আর একটু ভালো করে শিখতে পারতাম। জীবন কাজ সবকিছুই আর একটু অন্যভাবে সমাহৃত করার চেষ্টা করা যেত।

কেউ যদি সেই যাদুটায় পৌঁছনোর চেষ্টা করেন, আমার চেয়ে অনেক তরুণ, এই লাগ-এর বেশির ভাগ ছেলে যেমন, অনেক জীবন্ত, এদের মতোই আরো অন্য নতুন কেউ, তাদের যদি একটু কাজ এগিয়ে দেওয়া যায়। কারণ, এটা করতে তো আমারও অসম্ভব পরিশ্রম হচ্ছে, কলেজের কলিগরা অনেকেই বলছে, করে তোমার লাভ কী, তোমার লাইনও তো না, পয়সাও পাবেনা, তাহলে? নিজেরও মনে হয় মাঝে মাঝে। গত কাল আমার পা ভাঙল, মধ্যমগ্রাম প্ল্যাটফর্মের গায়েই লাইটপোস্ট ধরে বসে পড়তে পড়তেই মনে এল, যদি ভেঙে থাকে, প্লাস্টার করতে করতে কলেজ যেতে যেতে অন্তত দিন সাতেক, তার মধ্যে এক দুই নেমে যাবে। সকালে এক্সরে করে ডাক্তার দেখিয়ে এলাম, সত্যিই ভেঙেছে — আজ এক নেমে যাচ্ছে। এখন বিকেল। পরের অংশটার প্রথম ড্রাফ্ট লেখা আছে। রাতের মধ্যে নেমে যাবে। এই তাগিদটা সেই যাদুর। একটা কমিউনিটি। একটা স্বপ্ন। তার নাম গ্নু-লিনাক্স। আমিও তার অংশ — আপনি হবেন কিনা, আপনি ঠিক করবেন।

২।। অ্যাপ্লিকেশন, সফটওয়ার, অপারেটিং সিস্টেম — সবই প্রোগ্রাম

এতক্ষণ অব্দি হার্ডওয়ার যে উপাদানগুলো নিয়ে আমরা আলোচনা করলাম — এদের মেলালে সেটা হয় একটা কম্পিউটারের খোলস, কম্পিউটার নয়। কম্পিউটার হল আপনার সামনের সেই আস্ত পিসি-টা যাতে আপনি লেখেন, গান বাজান, অ্যাকাউন্ট রাখেন, সিনেমা দেখেন, অঙ্ক কষেন, প্রোগ্রাম লেখেন, কম্পাইল করেন, যা দিয়ে আপনি ইন্টারনেটে ওয়েবসাইট থেকে ওয়েবসাইট চেখে বেড়ান মানে ব্রাউজ করেন, মেল লেখেন দূরে থাকা আপনজনকে — সেই জ্যান্ত জিনিষটা হল কম্পিউটার। সেই যন্তরটা হল ওই হার্ডওয়ার এবং সেই হার্ডওয়ারকে মিলিয়ে সেই হার্ডওয়ারের উপর সক্রিয় একটা সফটওয়ার ব্যবস্থা।

ক্রমে ক্রমে আমরা আরো ভালো করে বুঝব, প্রথমে ভাবা যাক, এই সফটওয়ারটা কী? একদম সরাসরি একটা মেশিন থেকে শুরু করি আসুন। আপনি মেশিনটা অন করলেন। স্ক্রিনের উপর দিয়ে অনেক অনেক লাইন অক্ষর ভেসে গেল। এবং শেষ অব্দি ফুটে উঠল হয় একটা ছবি — তার মধ্যেও কিছু লেখা আছে, নয়তো নিতান্তই কয়েক লাইন লেখা। যদি এটা একটা উইনডোজ বাতাবরণ হয় তাহলে ফুটে উঠল উইনডোজ ৯৮ কি ২০০০ কি এক্সপি, এবং তার সঙ্গে তার লোগোর ছবিটা। আর গ্নু-লিনাক্স হলে সেখানে সম্ভাব্য অনেকগুলো রাস্তা আছে, এক, সেটা কনসোল মোড হতে পারে, মানে কালো স্ক্রিনের উপর সারি সারি অক্ষর, তাতে লেখা থাকবে 'ম্যানড্রেক ৯' কি 'সুজে ৮.২' কি 'ডেবিয়ান ৩' কি 'রেডহ্যাট ৯' কি এইরকম কিছু, বা, গ্নু-লিনাক্স-এ নিজের ইচ্ছেমত দুইরকম ব্যবস্থাই করা যায় — আপনি চাইলে এই কনসোল মোডে না ফুটে উঠে উঠবে ওই উইনডোজ-এর মতই, ছবিসহ, লোগোসহ — এর নাম গুই মোড (GUI — **G**raphical-**U**ser-**I**nterface)। পরে অনেক বিশদ ভাবে আলোচনা আছে গুই-এর, এখন এটুকু মাথায় রাখুন যে, কনসোল মোডে কাজ করতে হয় লিখিত আদেশ কম্পিউটারকে কিবোর্ড থেকে টাইপ করে, তারপর এন্টার চাবিটা মেরে, মানে সেই নির্দেশটাকে কম্পিউটারে এন্টার করিয়ে। এন্টার হল মানে আদেশটা কম্পিউটারে পৌঁছল, সে কাজ শুরু করল।

আর, কনসোল মোডে এই টাইপ করে কাজ করার একটা বিকল্প ব্যবস্থা তৈরি হয়েছিল, কম্পিউটার মানবসমাজে কাম মানবজীবনে আসার বেশ কিছু বছর পরে, যেখানে আদতে কাজটা হয় দুটো স্তরে। একটা আসল স্তর যা গোপন রয়ে যায় আপনার কাছে, ব্যবহারকারীর কাছে। প্রকাশ্য স্তরে আপনি দেখেন একটা ছবি, ছবির গায়ে সুইচ, সেই সুইচের গায়ে মাউস দিয়ে ক্লিক করলেই কাজ শুরু হয়। মাউসের সঙ্গে কখনো কখনো অবশ্য ফাইলের নাম, ডিরেক্টরির নাম জাতীয় ছোটখাট জিনিষ কিবোর্ড দিয়ে টাইপ করেও ঢোকাতে হয়। মানে, আদেশের মূল মাধ্যমটা এখানে আর কিবোর্ড নয়, এখানে আপনি আদেশ পাঠাচ্ছেন মাউস দিয়ে। এই প্রকাশ্য স্তরের নিচে কাজ করছে একটা অপ্রকাশ্য স্তর যেখানে আপনার এই মাউস-ইভেন্ট মানে ক্লিক করার ঘটনাটাকে কম্পিউটারের অপারেটিং সিস্টেম অনুবাদ করে নিচ্ছে কনসোল মোডের আদেশের মত অক্ষর এবং শব্দভিত্তিক কমান্ডে, এবং সেই কমান্ড অনুযায়ী কাজ করে চলেছে। যেরকম আদেশ আপনি নিজেই দিতেন যদি আপনি কনসোল মোডে বা কমান্ড মোডে কাজ করতেন। এখন আপনি তা করছেন, ছবি এবং মাউস দিয়ে এই কাজ করাটাই হল গুই মোড।

কনসোল হোক আর গুই হোক যেকোনো একটা মোডে আপনি কম্পিউটারে ঢুকলেন। এবার আপনি যে কাজটা চান সেটাকে আপনার চালু করতে হবে। ধরুন আমরা তিনটে কাজ ভাবি, লেখা, গান শোনা, এবং প্রোগ্রাম লেখা আর চালানো। প্রথম কাজটা ওয়ার্ড প্রসেসরের। গ্নু-লিনাক্স হলে এই প্রোগ্রামটার নাম ওপন-অফিস বা কে-অফিস বা অ্যাবিওয়ার্ড বা অন্য কিছু। লিনাক্সে সমস্যাটা দারিদ্রের নয়, প্রাচুর্যের। যে কোনো কাজের জন্যেই এত এত প্রোগ্রাম তৈরি হয়ে আছে যে প্রথম দিকে অনেক ব্যবহারকারীর কাছে সেটা একটা গোলকধাঁধার মত লাগে। প্রতিবেশটা উইনডোজ হলে, ওয়ার্ডপ্রসেসিং বা শব্দ-চটকানোর জন্যে আপনি ব্যবহার করছেন এমএস ওয়ার্ড বা অ্যাডোব পেজমেকার বা লোটাস স্মার্টশ্যুট রাইটার।

গান শোনার জন্যে আপনি গ্নু-লিনাক্সে ব্যবহার করতে পারেন এমপিজি১২৩ বা এমপিজি৩২১ বা ফ্রি-অ্যাম্প বা এক্সএমএমএস বা এমপ্লেয়ার বা এরকম অজস্র কিছুর একটা। উইনডোজ হলে যেখানে ব্যবহার করতেন উইনডোজ মিডিয়া প্লেয়ার বা উইনঅ্যাম্প।

আর প্রোগ্রাম লেখাটাও লেখা, কিন্তু তার জন্যে ওয়ার্ড প্রসেসরে কাজ হয়না, ওয়ার্ড প্রসেসর কাজ করে আপনার লেখার ফর্মাটিং নিয়ে। আর প্রোগ্রামগুলো লেখা হয় সাদামাটা বা প্লেন টেক্সটে, সেখানে কোনো ফর্মাটিং থাকেনা। ফন্ট বা অক্ষর কী হবে, প্যারাগ্রাফের আগে পরে ডানে বাঁয়ে জায়গা ছাড়া হবে কিনা, ইত্যাদি সবকিছুই এক কথায় হল ফর্মাটিং। শূন্য নম্বর দিনের ফর্মাটিং নিয়ে আলোচনা মনে করুন। ফর্মাটিংরিক্ত নিখাদ নির্জলা টেক্সট লেখার এডিটরগুলোকে সচরাচর ডাকা হয় কমান্ড এডিটর বলে। গ্নু-লিনাক্সে কমান্ড এডিটর আছে ইম্যাক্স ভিম জো জেডিট নেডিট গেডিট কে-রাইটার জাতীয় গুচ্ছ গুচ্ছ সফটওয়ার। প্রোগ্রামিং করা মানে কিন্তু শুধু প্রোগ্রাম লেখা না। প্রোগ্রাম লেখা শেষ হওয়ার পরে তাকে কম্পাইল করতে হয় কম্পাইলার দিয়ে। এগুলোর সঠিক অর্থ কী তাই নিয়ে এখুনি মাথা ঘামাবেন না, সময় হলেই আপনি জেনে যাবেন, জানতে হবে আপনাকে। এই কম্পাইল করার কাজের জন্যে গ্নু-লিনাক্সে আছে গ্নু-কম্পাইলার-কালেকশন বা জিসিসি। বলতে মাত্র একটা বাক্য লাগল, কিন্তু এটাই আসলে গোটা গ্নু-লিনাক্স আন্দোলনের নিষেকভূমি, বিরাট একটা ইতিহাস আছে এখানে, যার একটু আমরা জানব পাঁচ নম্বর দিনের শুরুতে। আর উইনডোজ হলে লেখার জন্যে কমান্ড এডিটর হিশেবে নোটপ্যাড আর কম্পাইল করার জন্যে বোরল্যান্ড মাইক্রোসফট বা ইনটেল ইত্যাদি কম্পাইলার, নানা প্রোগ্রামিং ভাষায় লেখা প্রোগ্রাম কম্পাইল করার।

প্রোগ্রাম লেখা আর কম্পাইল করার আর সেই কম্পাইল হওয়া প্রোগ্রাম চালানোর কাজগুলোকে আলাদা আলাদা ভাবে না করে, একটা অ্যাপ্লিকেশনের মধ্যেও করার ব্যবস্থা আছে। তাকে বলে আইডিই (IDE — **I**ntegrated-**D**evelopment-**E**nvironment) গোছের জিনিষগুলোকে আনলাম না। উইনডোজ বা গ্নু-লিনাক্স দু ধরনের ব্যবস্থাতেই আছে। সেগুলোও আপনি ব্যবহার করতে পারেন প্রোগ্রামিং এর কাজে। যাই হোক, এমন অবস্থা ভাবুন, যেখানে, গ্নু-লিনাক্স হোক বা উইনডোজ আপনি এই তিনটে কাজ করছেন, ওয়ার্ড-প্রসেসিং, গান-শোনা, এবং প্রোগ্রামিং। এই তিনটে কাজ আপনি চালাচ্ছেন আপনার মেশিনে, আপনার সিস্টেমে। এর পুরো প্রক্রিয়াটাকে একটু বোঝার চেষ্টা করা যাক। আপনি এই সময়ে কী কী করছেন, এবং আপনার মেশিনই বা কী কী করছে।

ওয়ার্ড প্রসেসর মানে ফর্মাটিং সহ লেখার জন্যে আপনি যে প্রোগ্রামই ব্যবহার করুন, যে বাতাবরণই হোক, উইনডোজ বা গ্নু-লিনাক্স — এবার আপনাকে টাইপ করে নিজের লেখাটাকে লিখে যেতে হবে এবং প্রয়োজন মোতাবেক মাউস বা কিবোর্ড ব্যবহার করে সেখানে নিজের ইচ্ছামত কাজ করতে হবে, বদল আনতে হবে। ধরুন, পরপর তিনটে লাইনের মোট লেখাটা আপনি গাবদা গাবদা গরুপাঠ্য সাইজের বর্ণমালায় দেখতে চান, তার জন্যে আপনাকে মাউস দিয়ে বা কিবোর্ড দিয়ে গোটাটাকে সিলেক্ট করতে হবে, এবং মাউস বা কিবোর্ড দিয়ে আদেশ দিয়ে তাদের ফন্ট সাইজ বাড়াতে হবে। আবার একটা শব্দ যোগ করতে হলে সেটা করতে হবে কিবোর্ড দিয়ে। খুব বিশেষ রকমে এমনকি সেই কাজটাও যদিও মাউস দিয়েই করা যায়। সেই রকম অ্যাপ্লিকেশন আছে যা স্ক্রিনে একটা কিবোর্ডের ছবি ফুটিয়ে তোলে, এবং সেখানে যে অক্ষরের ছবিতে আপনি ক্লিক করবেন, আপনার লেখায় সেটা যোগ হয়ে যাবে। বা গান শুনতে হলে আপনাকে গানের প্রোগ্রামটা খুলে কোন গান আপনি বাজাতে চান সেই ফাইলটাকে তার ডাইরেক্টরিতে আপনাকে খুঁজে দিতে হবে। সেটাও করবেন মাউস বা কিবোর্ড দিয়ে। গানটা বাজতে শুরু করবে। বা কম্পাইল করছেন যখন প্রথমে আপনাকে প্রোগ্রামটা লিখে সেভ করতে হবে একটা ফাইলনাম দিয়ে, তারপর

তাকে কম্পাইল করতে বলতে হবে। তার মানেও সেই মাউস আর কিবোর্ড। কম্পাইল করা মানে কী সেটা আমরা একটু বাদেই জানব, উতলা হবেন না।

এই প্রতিটি ক্ষেত্রেই আপনি কী করছেন — হয় আপনার ইচ্ছের প্রোগ্রামের নাম লেখা ছবি দেওয়া আইকনে ক্লিক করছেন, মানে মাউস দিয়ে আদেশ দিচ্ছেন, প্রোগ্রামটা চালু হচ্ছে, অথবা কিবোর্ড থেকে আদেশ দিচ্ছেন টাইপ করে। কী আদেশ দিচ্ছেন? অমুক নামের প্রোগ্রামটা চালু করো। এই প্রোগ্রামটা হল একটা প্রয়োগ — অ্যাপ্লিকেশন। এরা কম্পিউটারের গোটা ব্যবস্থার একদম উপরের স্তর। যে কোনো একজন ব্যবহারকারী সরাসরি মুখোমুখি হয় এই অ্যাপ্লিকেশনগুলোর। এই অ্যাপ্লিকেশনগুলো কিছু ক্ষেত্রে একটা একক প্রোগ্রাম হতে পারে, অনেকক্ষেত্রেই একাধিক প্রোগ্রামের একটা সমাহার। যেমন ওপন-অফিস হোক বা এমএসঅফিস, লিখতে লিখতে আপনি একটা ছবি ঢোকাতে চাইলেন বা একটা সমীকরণ, ফন্ট মানে অক্ষরের চেহারা বদলাতে চাইলেন, বা সরাসরি এইখান থেকেই একটা মেল করতে চাইলেন কাউকে — এই প্রত্যেকটা কাজের সময়েই পর্দার পিছনে মূল প্রোগ্রামটা, মানে ওপন-অফিস বা এমএসঅফিস, ডেকে নিল নানা ধরনের নানা প্রোগ্রামকে। এই সমস্ত প্রোগ্রামগুলোকে মিলিয়েই হল গোটা প্রয়োগ বা অ্যাপ্লিকেশনটা।

এবার ভাবুন এই প্রোগ্রামগুলো কী? এগুলো আসলে ফাইল। সেখানে পরপর লেখা আছে আদেশ। কিন্তু আদেশগুলো এমন ভাষায় লেখা যা আপনি আমি বুঝব না, কিন্তু কম্পিউটার বুঝবে। যার নাম মেশিন ল্যাংগোয়েজ। দিন শূন্যর সেই ভিশুয়াল চ্যাংড়ামোটা মনে করুন, যেখানে আমরা একটা পিডিএফ মানে বিশেষ ফর্মাটিং-এর ফাইলকে কোনো একটা কমান্ড এডিটর দিয়ে খুলেছিলাম। নিছক ফাইলটাই কিন্তু প্রোগ্রাম নয়, বা যে কোনো ফাইলও প্রোগ্রাম নয়। সেই সব ফাইলেরাই প্রোগ্রাম যারা চলতে পারে। সেই ফাইল যা নিজে চলতে পারে, সে যখন চলে সেটা হল ওই নামের প্রোগ্রামের একটা প্রসেস বা প্রক্রিয়া। এগুলো সবই খুঁটিয়ে খুঁটিয়ে বুঝতে হবে আমাদের। নানা রকম জটিলতা নিয়ে ধীরে ধীরে। এখন শুধু একটা প্রাথমিক ধারণা করে নিচ্ছি আমরা। যা বলছিলাম, সব ফাইল নিজে চলতে পারেনা। ধরুন, আপনার ওয়ার্ডপ্রসেসরে আপনি একটা প্রবন্ধ লিখলেন, প্রবন্ধটা নিজের পছন্দমতন একটা নাম দিয়ে সেভ করলেন মানে তুলে রাখলেন কম্পিউটারের কোনো ডিস্কে, কোনো স্মৃতির ভাঁড়ারে। এবার হার্ডডিস্কিত ওই প্রবন্ধটা একটা ফাইল, কিন্তু নিজে নিজে চলেনা। আবার আপনার ওয়ার্ড-প্রসেসরটাও একটা ফাইল যা নিজে নিজে চলে। নিজে নিজেও চলে, আবার ওই প্রবন্ধের নাম করে আদেশ দিলে প্রবন্ধটাও খুলে দিতে পারে, তাকে আপনি আর একবার ওয়ার্ডপ্রসেস করতে পারেন, শব্দ-চটকাতে পারেন। এই আদেশটা আপনি মাউস দিয়েও দিয়ে থাকতে পারেন, ফাইল মেনুতে গিয়ে ওপন সাবমেনু ক্লিক করলে সে হার্ডডিস্কের ভিতর ফাইল আর ডিরেক্টরির কাঠামোটা খুলে দেয়, সেখানে প্রবন্ধটা খুঁজে দিতে হয়। আবার সরাসরি কিবোর্ড দিয়েও দেওয়া যেত আদেশটা।

এবার নিজে নিজে চলা এই ফাইলগুলো মানে প্রোগ্রামগুলোকে মিলিয়ে মিলিয়েই তৈরি হয় অ্যাপ্লিকেশন। এই চলমান প্রোগ্রাম-সমাহার অ্যাপ্লিকেশনগুলোর বন্যের বন এবং শিশুর মাতৃক্রোড় হল আপনার কম্পিউটারের সামগ্রিক প্রতিবেশটা, মানে আপনার মেশিনের অপারেটিং সিস্টেম। ধরুন ম্যানড্রেক ৯.১ কি সুজে ৮.২ কি উইনডোজ ২০০০ — এরা হল অপারেটিং সিস্টেম। এরকম আরো আছে ইউনিক্স, সোলারিস, আইক্স ইত্যাদি। এই অপারেটিং সিস্টেমও এক রকমের প্রোগ্রাম। তার মানে একটা ফাইল বা একাধিক ফাইলের একটা সমাহার। যে ফাইল নিজে চলে এবং অন্য সব প্রোগ্রামকে চালায়। অপারেটিং সিস্টেম নামক প্রোগ্রামটা আবার শুধু অন্য ফাইলদেরই চালায় না, চালায় হার্ডওয়ারগুলোকেও, অন্য প্রোগ্রামদের হার্ডওয়ার ব্যবহারের ব্যবস্থা করে দেয়। একটা চলমান কম্পিউটারে সবচেয়ে প্রাথমিক, সবচেয়ে শক্তিশালী এবং সবচেয়ে জরুরি প্রোগ্রাম এই অপারেটিং সিস্টেম। আমরা ক্রমে আরো বিশদ জানব এদের নিয়ে।

২.১।। ভৌত উপাদান — কম্পিউটার সিস্টেমের ভিত

তাহলে একটা জ্যান্ত কম্পিউটারের উপাদান কী কী? একদিকে গুচ্ছ গুচ্ছ হার্ডওয়ার। সিপিইউ, র‍্যাম, রম, হার্ডডিস্ক, ফ্লপি ইত্যাদি। এর সঙ্গে আছে গোটা প্রোগ্রাম জগতটা — অপারেটিং সিস্টেম থেকে অ্যাপ্লিকেশন। এরা আবার কাজ করে ওই হার্ডওয়ার উপাদানগুলোর উপরেই, এদের ব্যবহার করে, এদের উপর, এদের দিয়ে। এদের এক এক পিস এক এক জাতের জিনিষ, কারুর লেজ নেই, কারুর বা উনিশটা লেজ, কেউ আবার লেজ কাকে বলে তাই বোঝে না।

তাহলে এই বহুমুখী বিচিত্রমুখী এইসব বিদঘুটেপনাকে একত্রিত করে মিলিয়ে একটা চলমান কম্পিউটারের নিখুঁত সমগ্রে নিয়ে আসছে কে?

এই সমস্ত উপাদানকে যথাযথভাবে নিয়ন্ত্রণ এবং ব্যবহার করার কাজটা কিন্তু আদৌ সহজ না। ইন ফ্যাক্ট, সাইবার জগতে এটাই সবচেয়ে খিঁটকেল কাজ। এটাই সেই ভিত, যার উপর দাঁড়িয়ে রয়েছে আর সব কিছু, সমস্ত কিছু। এই কাজটা সহজ নয়, তাই এই কাজ করার ক্যালিসম্পন্ন প্রোগ্রাম লেখার কাজটাও একটা সহজ নয়। জটিলতার একটা খুব দুধেভাতে জাতের আন্দাজ নেওয়ার চেষ্টা করা যাক। ধরুন আপনি আপনার ওয়ার্ড প্রসেসরে যে প্রবন্ধটা লিখছেন, সেটায় কিছু বদল আনলেন, এনে একটা অন্য নামে সেভ করলেন। তারপর তার একটা প্রিন্ট-আউট নিলেন। এখানে অপারেটিং সিস্টেমের কাজের একটা আলগা এবং উপর-উপর তালিকা বানানো যাক।

এর মানে আপনার আদেশ অনুযায়ী ওয়ার্ড প্রসেসর নামক প্রোগ্রামটা ব্যবহার করল সিপিইউকে, যে মূল কাজটা করেছে। সিপিইউ যা করে, কম্পিউটারে কেন্দ্রীয় ভূমিকাটাই তার, সব কাজই তার। তাকে কাজে লাগাল প্রোগ্রামটা। তার মানে, এই প্রোগ্রামের হয়ে সিপিইউকে কাজে লাগাল অপারেটিং সিস্টেম বা কারনেল। আগেই তো বলেছি, এই হার্ডওয়ার উপাদানগুলোর সরাসরি নিয়ন্ত্রণ একমাত্র সব প্রোগ্রামের নিয়ন্তা প্রোগ্রাম মানে কারনেলের, সাধারণ প্রোগ্রামের কোনো রাইটই নেই হার্ডওয়ারের সঙ্গে সরাসরি কথা বলার।

শুধু সিপিইউ নয় কিন্তু, কাজে লাগানো হয়েছে মেমরিকেও। অন্য নামে ফাইলটাকে সেভ করার আগে মেমরির আলমারির তাকে তুলে নিতে হয়েছে গোটা ফাইলের মোট তথ্যটাকে। ওই প্রবন্ধের ফাইলটার অন্তর্বস্তু নিয়ে কিছু করা মানেই তো ফাইলের আভ্যন্তরীন তথ্যটাকে তুলতে তুলতে নামাতে নামাতে যাওয়া। মেমরিকে কাজ করতে হচ্ছে নিরন্তরই। হার্ডডিস্কে হোক কি যেখানে হোক এক বাইট তথ্য নামানো বা ওঠানো মানেই, আগেই সেই বাইটটাকে মেমরিতে তুলে নেওয়া।

একাধিক বার একাধিক ভাবে কাজে লাগাতে হয়েছে হার্ডডিস্ককে। প্রথমে, হার্ডডিস্কের ভিতরের সেই থালা বা প্ল্যাটারগুলোকে ঘুরিয়ে ঘুরিয়ে সেই সেই সেক্টরগুলো আনতে হয়েছে যেখানে মূল ফাইলটা লেখা ছিল, তাদের শরীরে লিখে রাখা তথ্যটাকে পড়ে তুলে নিতে হয়েছে মেমরির তাকে, তারপর নতুন নামে সেভ করার সময় আবার একবার চরকিপাক ঘোরাতে হয়েছে বেচারা হার্ডডিস্ককে, এবার তার শরীরে আবার কিছু নির্দিষ্ট সেক্টরে লেখা হয়েছে নতুন কিছু তথ্য — মানে নতুন নামের তথ্য। এর সঙ্গে সঙ্গে অগোচরে চলেছে আরো কিছু কাজ, হার্ডডিস্কের পার্টিশনে কী কী ডিরেক্টরিতে কী কী ফাইল বদলালো তার বদলে যাওয়া খুঁটিনাটি আবার লিখে রাখতে হয়েছে হার্ডডিস্কের মধ্যেই লিখে রাখা ফাইল-সংক্রান্ত তথ্যে, যার নাম মেটাডেটা। ডেটার বিষয়ে ডেটা। ভালো করে এটা বুঝব আট নম্বর দিনে গিয়ে।

কাজে লাগাতে হয়েছে আপনার কনসোলকে, যখন আপনি ফাইলকে বদলেছেন, ফাইলের প্রত্যেকটা বদলকে সে স্ক্রিনে ফুটিয়ে তুলেছে। আপনার প্রোগ্রাম অ্যাপিল করেছে অপারেটিং সিস্টেম বা কারনেলের কাছে, কারনেল কথা বলেছে স্ক্রিন নামক ভৌত উপাদানের কন্ট্রোলারের সঙ্গে, কন্ট্রোলার সরাসরি তথ্য পাঠিয়েছে স্ক্রিন বা কনসোলকে।

কাজে লাগানো হয়েছে প্রিন্টারকে, যখন সে প্রিন্ট নিয়েছে। আবার ওই প্রোগ্রাম থেকে কারনেল থেকে প্রিন্টার-ড্রাইভার থেকে প্রিন্টার-কন্ট্রোলার থেকে প্রিন্টার। ফটফটে সাদা কাগজে খটখটে কালো বর্ণমালায় ফুটে উঠেছে আপনার প্রবন্ধ। চাইলে ইস্টম্যানকালার প্রবন্ধও লিখতে পারেন, সরকারি কোনো নিষেধ নেই।

উপরের এই তালিকাটা কিন্তু অতিসরলীকৃত বললেও কম বলা হয়। কারণ, প্রতিটি মুহূর্তে আপনার চালানো প্রোগ্রামের পাঠানো আর্জি রাখতে গিয়ে কারনেলকে মাথায় রাখতে হয়েছে আপনার গোটা মেশিনের প্রতিটি হার্ডওয়ার ডিটেইলস, আপনার মাদারবোর্ডের সার্কিটে বিদ্যুৎপ্রবাহের পারমানবিক ওঠানামা, প্রতিটি আইসির পিনের সংখ্যা, মেশিনের হার্ডডিস্কের প্রতিটি প্ল্যাটারের প্রতিটি ডিটেইলস — উঃ, চাইলে এই তালিকাটাকে আরো বহু সময় ধরে বাড়িয়ে চলা যায়, প্রায় অ্যাড ইনফিনিটাম।

এই গোটা ঘটনাটা ঘটল কী করে? কে এই গোটা বহুমুখী প্রক্রিয়াটাকে নিয়ন্ত্রণ করল। আপনার ওয়ার্ড প্রসেসর প্রোগ্রামের ভূমিকাটুকু খেয়াল করুন। তার আর্জি মোতাবেক ঘটেছে এই প্রতিটি ঘটনা। ওয়ার্ড প্রসেসর প্রোগ্রামটার ভিতরেই সেই ব্যবস্থা করা ছিল। তার শরীরে ইঁদুর ছেড়ে দিয়ে বা কিবোর্ড ঠুকে আপনি যখনি কোনো আব্দার

জানাবেন, তার সঙ্গে তাল মিলিয়ে ঠিক কী আর্জি জানাতে হবে কারনেলের কাছে। যখনই আপনি প্রিন্ট করতে বলবেন, আপনার ওয়ার্ড প্রসেসর জানে, কোন জায়গায় কোন ভাবে কী আবেদন জানাতে হবে, বাবা কারনেলের চরণে সেবা লাগি, এই এই তথ্য এই এই দিকে, প্রিন্টারের পোর্টের দিকে পাঠিয়ে দাও বাবা, যাতে সেটা কেবল বেয়ে কেবল প্রিন্টারের কাছেই পৌঁছয়, এবং আপনি ছাপার আখরে নিজের লেখা পড়ে মুগ্ধ হতে পারেন।

অর্থাৎ হার্ডওয়ারের উপাদানগুলোকে তার কাজে লাগাতেই হবে। শুধু তার কেন, যে কোনো প্রোগ্রামেরই। প্রোগ্রামগুলোর কাজ কী? চলা, চলে কোনো একটা কাজ করা। কাজটা কোথায় হয়? হার্ডওয়ারে। এবার এই বিভিন্ন হার্ডওয়ার উপাদান, তাদের এক এক জনের এক এক রকম খুঁটিনাটি। কেউ তৈরি চৌম্বক পদার্থে, কেউ হল একটা পিকচার টিউব, কেউ আবার একটা নড়তে থাকা কাঁটা, পিছনে একটা কালির দোয়াত। তাদের প্রত্যেকটা খুঁটিনাটিকে মিলিয়েই তাকে কাজ করতে হবে। সেই কাজের সুবিধের জন্যে এই সমস্ত উপাদানের উপর সর্বব্যাপী একটা প্রলেপের মত ছড়ানো থাকে সফ্টওয়ারের একটা স্তর — এটারই সিরিয়াস নাম অপারেটিং সিস্টেম। প্রোগ্রামগুলোকে একটা সহজ ব্যবহারযোগ্যতা দেওয়াই যার কাজ। 'ফ্লপিতে লেখো', 'মোডেমে সংযোগ করো' বা 'হার্ড-ডিস্কের এই ফাইল উড়িয়ে দাও' — আপনি এই আদেশ দিলেই আপনার প্রোগ্রামের হয়ে এই অপারেটিং সিস্টেম চট করে বুঝে নিতে পারে 'ফ্লপি' বা 'মোডেম' বা 'হার্ড-ডিস্ক' কী ধরনের বস্তু কিম্বা 'লেখা' বা 'সংযোগ' বা 'ওড়ানো' মানেই বা কী।

এই তালিকায় আমরা কম্পিউটার হার্ডওয়ার থেকে সফটওয়ারের স্তরবিন্যাসটা দেখিয়েছি। শুধু একটা জিনিষ খেয়াল করার — স্তরগুলো কিন্তু নিচ থেকে উপরে। ভিত-এর উপর বাড়ির মত।

<table>
<tr><th colspan="4">হার্ডওয়ার থেকে সফটওয়ার স্তরবিন্যাস</th></tr>
<tr><td>(৬) হিসাবনিকাশের সফটওয়ার</td><td>(৬) রেলের টিকিট বিলিব্যবস্থা</td><td>(৬) ইন্টারনেটে ঘোরাঘুরি</td><td>প্রায়োগিক সফটওয়ার বা অ্যাপ্লিকেশনের এলাকা</td></tr>
<tr><td>(৫) কম্পাইলার</td><td>(৫) এডিটর</td><td>(৫) কমান্ড ইন্টারপ্রিটার</td><td rowspan="2">মূল কাঠামো বা সিস্টেম প্রোগ্রামের এলাকা</td></tr>
<tr><td colspan="3">(৪) অপারেটিং সিস্টেম</td></tr>
<tr><td colspan="3">(৩) মেশিন ল্যাংগুয়েজ</td><td rowspan="3">হার্ডওয়ার বা ভৌত যন্ত্রপাতির এলাকা</td></tr>
<tr><td colspan="3">(২) মাইক্রোআর্কিটেকচার</td></tr>
<tr><td colspan="3">(১) ভৌত উপাদানগুলো</td></tr>
</table>

এক নম্বর হল হার্ডওয়ার-এর এলাকা, এর মধ্যে তিনটে স্তর। ভৌত উপাদান, মাইক্রোআর্কিটেকচার, মেশিন ল্যাংগুয়েজ। এর পরের এলাকা মূল কাঠামো বা সিস্টেম প্রোগ্রামের — মেশিনের অপারেটিং সিস্টেম কী ভাবে চলে। এর মধ্যে স্তর আবার দুটো। এক নম্বরে একা অপারেটিং সিস্টেম মানে ৪ নম্বর। উপরেরটায় মানে পাঁচ নম্বর স্তরে তিনটে অংশ — কমান্ড ইন্টারপ্রিটার, এডিটর এবং কম্পাইলার। একদম উপরের স্তরটা মানে ৬ নম্বর হল প্রায়োগিক সফটওয়ার বা অ্যাপ্লিকেশন। এর মধ্যে আবার নানা বিচিত্র অংশ আছে। আমরা এখানে তিনটে উদাহরণ দিয়েছি। ছকটা মাথায় তুলে ফেলুন, ধীরে ধীরে প্রয়োজনীয় খুঁটিনাটিগুলো স্পষ্ট হয়ে যাবে পাঠমালাটা চলতে চলতে। আপনাকে মাঝে মাঝেই ফেরত আসতে হবে পুরোনো দিনগুলোয়। এখানে স্তর ভাগ ছটা — একদম প্রাথমিক বা নিচের স্তরটা হল ভৌত উপাদানের মানে সার্কিট প্লাস্টিক ধাতু সিলিকনের। আপনার পিসির টাওয়ারের গায়ের ক্যাবিনেটটা খুললে মধ্যে আপনি যে জগতটা পাবেন। মাদারবোর্ড, এক্সপ্যানশন স্লট, এক্সপ্যানশন কার্ড, সার্কিট বোর্ড, অ্যাডাপ্টার, ট্রানজিস্টর, ক্যাপাসিটর, তার, তামা, রেজিস্টান্স, কানেক্টর, হার্ডডিস্ক-এর কৌটো, সিডি-ড্রাইভের কৌটো, ফ্লপি-ড্রাইভের কৌটো, দু-চারটে ফ্যান — এটা মূলত ইলেকট্রিকাল আর ইলেকট্রনিক এঞ্জিনিয়রের এলাকা। এই ১ নম্বর ভৌত উপাদানের স্তরের ঠিক উপরে উপরে দাঁড়িয়ে আছে মাইক্রোআর্কিটেকচার আর মেশিন ল্যাংগুয়েজের ২ আর ৩ নম্বর স্তর। এই ১, ২ আর ৩ নম্বর স্তরকে মিলিয়ে হার্ডওয়ার বা ভৌত উপাদানের এলাকা। দেখুন আমরা ডানদিকে দেখিয়েছি। এর উপরের এলাকাটা হল মূল কাঠামোটাকে ঘিরে প্রোগ্রামের বা সিস্টেম প্রোগ্রামের এলাকা, দেখুন ডানদিকে দেখানো।

তার মধ্যে আছে দুটো স্তর। ৪ নম্বর গোটা স্তরটা জুড়ে একা কুম্ভ অপারেটিং সিস্টেম, মাদার মাস্তান অফ অল মাস্তানস। তার উপরে ৫ নম্বর স্তরটায় আমরা তিনটে উপাদান দেখিয়েছি — কম্পাইলার এডিটর এবং কমান্ড ইন্টারপ্রিটার। এডিটর দিয়ে প্রোগ্রাম লেখা হয়, বা কনফিগারেশন টেক্সট ফাইল। এই এডিটর হল বাহুল্যবর্জিত ফর্মাটিং বর্জিত টেক্সট এডিটর, দেখানোর আর খাওয়ার দুটো আলাদা নয়, একটাই মাত্র দাঁত। কম্পাইলার দিয়ে লেখা প্রোগ্রামকে কম্পাইল করে চালানোর মত ফাইল বা এক্সিকিউটেবল প্রোগ্রাম ফাইল বানানো হয়। আর কমান্ড ইন্টারপ্রিটার হল আমাদের আর কম্পিউটারের মধ্যে দোভাষি। কোনো চালু প্রোগ্রাম মারফত বা সরাসরি সিস্টেমকে কোনো আদেশ যখন দিই আমরা, সেটার জন্যে সিস্টেমকে ঠিক কী করতে হবে — এটাই সিস্টেমকে বুঝিয়ে দেয় কমান্ড ইন্টারপ্রিটার। গ্নু-লিনাক্সে সবচেয়ে জনপ্রিয় কমান্ড ইন্টারপ্রিটার হল ব্যাশ। পরে এই নিয়ে গুষ্টি গুষ্টি পড়তে হবে আমাদের। আর এই ৫ নম্বর স্তরের উপরের গোটাটাই হল ব্যবহারকারীর ব্যবহৃত প্রায়োগিক বা অ্যাপ্লিকেশন সফটওয়ারের এলাকা — যে ৬ নম্বর স্তরে আমরা তিনটে প্রয়োগ সফটওয়ারের উদাহরণ দেখিয়েছি, হিশেবনিকেশ, রেলের টিকিট আর ইন্টারনেট ব্রাউজিং। উদাহরণ কিন্তু অজস্র হতে পারে। আমাদের চারপাশে সফটওয়ার বলতে আমরা সচরাচর যা বুঝি তার প্রায় গোটাটাই এই ৬ নম্বর স্তরের মধ্যে সীমাবদ্ধ।

এই স্তরের মধ্যে তিনটে তার উপরেরটা হল মূল কাঠামো বা সিস্টেম প্রোগ্রামের এলাকা। তার মধ্যে দুটো উপস্তর। একটা উপস্তর অপারেটিং সিস্টেমের একার। অন্যটা আর একদম উপরেরটা হল প্রায়োগিক সফটওয়ারের, মানে বেশিরভাগ ব্যবহারকারীর, এখানে তিনটে উদাহরণ দেওয়া হয়েছে। আরো কোটি কোটি আছে। সবচেয়ে বেশি কম্পিউটার এখন যেভাবে ব্যবহার হয় — বাজনা-টিভি-ব্রাউজারের থ্রি-ইন-ওয়ান — সেটাও এই ছয় নম্বর স্তরে। এবার আজকের আলোচনার অবশিষ্টটা জুড়ে আমরা কম্পিউটারের সফটওয়ারের এই এলাকাগুলো একটু ভালো করে বুঝব।

২.২।। মাইক্রোআর্কিটেকচার আর মেশিন ল্যাংগুয়েজ — হার্ডওয়ার থেকে সফটওয়ার

সিলিকন প্লাস্টিক ধাতু ইত্যাদি দিয়ে তৈরি ভৌত উপাদানগুলোর স্তরের ঠিক উপরের দু-নম্বর স্তরটাই হল মাইক্রোআর্কিটেকচারের। এখানে একদম কাঁচা আছোলা ভৌত উপাদানগুলোকে মিলিয়ে মিলিয়ে এক একটা কেজো ইউনিট বানানো হয়। এর কেন্দ্রীয় জায়গায় দাঁড়িয়ে অর্কেস্ট্রার কন্ডাক্টরের মত ব্যাটন দোলায় সিপিইউ চিপ। সিপিইউ মানে মূল চিপ এবং তার শরীরে দেগে দেওয়া ওই রেজিস্টারগুলো, একটু আগে যাদের কথা আমরা বললাম।

সিপিইউ-র ভিতর থাকে নানা ভৌত উপাদান — থাকে আলু মানে অ্যারিথমেটিক লজিক ইউনিট এবং অন্যান্য অংশ। এবং থাকেওই আলুর ভিতরে বাইরে অবশিষ্ট সিপিইউর সঙ্গে তার তথ্য বা ডেটা চলাচলের সংযোগ বা বাস। প্রতি নির্দিষ্ট সময় এককেকে নির্ভুল ভাবে নিরবচ্ছিন্ন ভাবে সূচিত করে চলার জন্যে, যাকে কম্পিউটার বিজ্ঞানে ডাকা হয় ক্লক-সাইকেল বলে, প্রয়োজনীয় সার্কিটও থাকে ওই সিপিইউ-র ভিতরেই, আমরা আগেই বলেছি।

সিপিইউ-র ওই রেজিস্টারগুলো থেকে সূচক তোলা হয় আর অ্যারিথমেটিক-লজিক ইউনিটে তাদের পাক দেওয়া হয় — যেমন, দুটো সংখ্যাকে যোগ করা বা বুলিয়ান অ্যান্ড (AND)। ফলাফলটা আবার তুলে দেওয়া হয় ওই রেজিস্টারের শিকেয়। ধরা যাক একটা হার্ড ডিস্কের একটা নির্দিষ্ট সেক্টর থেকে একটা সংখ্যাকে পড়া হল, আর একটা সেক্টর থেকে আর একটা সংখ্যা, তাদের যোগ করা হল, মানে ওই অ্যান্ড, তারপর সেই যোগফলটাকে ফের দেগে দেওয়া হল হার্ড ডিস্কের গায়ের আর একটা সেক্টরে। বাস্তবে জাটিল্যের সংখ্যা আর একটু বেশি। আরো অজস্র নিয়ন্তা বা প্যারামিটার তাদের নিজের নিজের খেলার প্রতিভা দেখিয়ে থাকে এখানে, যার কিছু ইঙ্গিত আমরা ইতিমধ্যেই দিয়েছি। কিন্তু, মাইন্ড ইট, আমি আগেই বলেছি, এও অতি অল্প হইল। এবং ওই ড্রাইভের মানে হার্ড ডিস্কের ব্যাপারটাও রীতিমত খ্যাঁচালো। আর তথ্য বা ডেটা ঢোকানো-বেরোনোর মানে ইনপুট-আউটপুটের কায়দাও শুধু ওই সবেধন হার্ডমণি নয়, তাও তো জানেন। ধরুন আলুতে এই যোগ বিয়োগ কষাকষির ফলাফলটা শেষ অব্দি যাবে মোডেম মারফত সার্ভার। সার্ভার বলতে বোঝায়, শিশুপাঠ্য রকমে বললে, জংশন স্টেশনের মত বহুমুখী যোগাযোগসম্পন্ন কিঞ্চিৎ গাবদা গতরের একটা কম্পিউটার। এবার ভাবুন, কষাকষির আর লেখালেখির মধ্যে সময়সীমাটা যদি খুব বেড়ে যায় তাহলে মধ্যের অংশগুলো বেয়ে মোডেম বেয়ে সার্ভার অব্দি পৌঁছনোর আগেই, সার্ভারটাই হয়ত যোগাযোগের পথটা বন্ধ করে দেবে। তার মানে গোটা কাজটাই মিনিংলেস হয়ে যাচ্ছে। এই রকম অজস্র জটিলতা থাকে সেখানে। এর উপরের, সিস্টেম প্রোগ্রামিং-এর এলাকার থেকে অপারেটিং সিস্টেম যখন

কোনো কাজ করিয়ে নিচ্ছে সিপিইউকে দিয়ে, সেই সিপিইউ কিন্তু কাজ করছে তার চারপাশের এই ভৌত জটিলতাকে নিয়েই। তাই অপারেটিং সিস্টেম তথা গোটা সিস্টেম প্রোগ্রামিং-কেই খেয়াল রাখতে হয়, মাথায় রাখতে হয় এই জটিলতাগুলোকে। কোন কাজ করাতে হলে ভৌত উপাদানগুলোকে কী করতে হবে। সেখানে কতগুলো জিনিষ সম্ভব, কতগুলো জিনিষ অসম্ভব, ইত্যাদি। ভৌত উপাদানের জগতের কাছে কোনো অন্যায় আব্দার জানালে চলবে না, সে ঠিক ততটুকুই পারবে, যতটুকু তার আসে। তার আসার চূড়ান্ত সীমায় তাকে ঠেলে দিয়েও যদি অপারেটিং সিস্টেমের কাজ না-চলে, তখন ভৌত উপাদানগুলোকেই বদলে ফেলার কথা ভাবতে হবে। যেভাবে ভৌত উপাদানের প্রকৌশলগত বদলগুলো ঘটে আর কি।

অনেক মেশিনে এই ভৌত উপাদানদের নিজস্ব জগতের এই নিজস্ব ঘ্যাঁটটা পাকানো হয় সফটওয়ার দিয়ে — তাদের বলে মাইক্রোপ্রোগ্রাম। অন্য অনেক মেশিনে আবার সেই একই কাজটা করা হয় হার্ডওয়ার সার্কিট দিয়ে। এই নিয়ে, কোনো রকম কোনো আলোচনাই আমাদের এই পাঠমালায় আসবে না। উঃ, এটা যে কী রিলিফ, হাফ জানা, কোয়ার্টার জানা, জানি-কি-জানিনা তাই ভালো বুঝছি না — এই সব ঘনচক্কর এলাকা নিয়ে কথা বলে চলাটা তবু সম্ভব, কিন্তু যে এলাকাটা স্পষ্ট জানি যে জানিনা, তাই নিয়ে যদি কথা বলতে হয়?

৩।। অপারেটিং সিস্টেম

যাইহোক, হার্ডওয়ারের যে জটিলতার কথা বলছিলাম। এই জটিলতাটা ফেস করার ফ্রাস্ট্রেশন থেকে পরিত্রাণায় জন্যে ধরায় অবতীর্ণ হয়েছিলেন অপারেটিং সিস্টেমের অবতার। খুব সহজ একটা উদাহরণ নেওয়া যায়। একটা ফ্লপিতে কী কী ফাইল সেটা আমরা জানি কী করে? উইন্ডোজে, এক্সপ্লোরার খুলে, ছবিটা যখন ফুটে উঠল, দুদিকে দুটো পাল্লা — বাঁদিকের পাল্লায় 'এ-ড্রাইভ'-এর গায়ে ক্লিক-মাত্র টাওয়ারের ফ্লপি ড্রাইভের চৌকো দরজাটার গায়ে ছোট্ট আলোটা জ্বলে ওঠে, দু-এক মুহূর্ত পরে ডানদিকের পাল্লায় ফুটে ওঠে সারি সারি ডাইরেক্টরি আর ফাইল। আর গ্নু-লিনাক্সে, এটা গুই হলে, ওই একই রকম ক্লিক করতে হয় '/mnt/floppy'-র গায়ে, একই ভাবে সারি সারি ফোল্ডার আর ফাইল উন্মুক্ত হয়ে যায়। লিনাক্সে অন্য আরো অনেকগুলো সম্ভাবনা থাকে — যাদের নিয়ন্ত্রণ করতে পারেন আপনি নিজেই। হতে পারে যে আপনার ফ্লপিতে মাউন্ট করতে হবে এই ব্যবস্থা আপনি করে রেখেছেন, মাউন্ট করা মানে কী তা আমরা অনেক পরে জানব, অন্তত প্রথম এগারো দিনে কোনো আশা নেই, তখন মাউন্ট করার আগে ক্লিক করলে আপনাকে কিছুই দেখাবে না। হতে পারে যে আপনি কনসোল মোডে কাজ করছেন, তখন আপনি কিবোর্ড থেকে টাইপ করে কমান্ড দিচ্ছেন — '`ls /mnt/floppy`' — মানে, '/mnt/floppy' ডিরেক্টরিতে মাউন্ট করা ফ্লপির ভিতর ফাইল-ডিরেক্টরির তালিকাটা দেখাও। বা, উইনডোজ হলে, মাউস-ক্লিক করার বদলে ডস-প্রম্পট-এ '`dir a:`' কমান্ড দিতে হত। উইনডোজ প্রতিবেশ মানেই মাউস, তা কিন্তু নয়, কনসোল মোডেও কাজ করা যায় বৈকি, এমএসডস (MS-DOS — **M**icro**S**oft-**D**isk-**O**perating-**S**ystem) প্রম্পট বলে একটা জিনিষ দেওয়া থাকে স্টার্ট — প্রোগ্রাম — অ্যাকসেসরিজ মেনুতেই, কিন্তু তাতে লিটারালি খাজনার চেয়ে বাজনা বেশি। কেন, সেটা সম্পূর্ণ অন্য আলোচনা — আমাদের জিএলটি ইশকুলের পাঠমালায় পড়েনা।

এবার, প্রথম ক্লিকের পর খুলে যাওয়া ডিরেক্টরি আর ফাইলের তালিকায় এবার আপনি ফের ক্লিক করেন আপনার পছন্দসই একটা ফাইলের গায়ে। এক মুহূর্তে খুলে যায় ফাইলটা। যদি টেক্সট ফাইল হয় খুলে যায় একটা টেক্সট সফটওয়ার অ্যাপ্লিকেশনের ভিতর। মানে একটা টেক্সট অ্যাপ্লিকেশন খোলে, তার ভিতরে আপনি দেখেন ফাইলটা খোলা আছে, আপনি পড়তে শুরু করেন। আবার যদি ওটা একটা গানের ফাইল হয়, মানে একটা এমপিথ্রি বা ওই গোছের ফাইল, তাহলে আপনি ক্লিক করা মাত্র খুলে যায় একটা গান-শোনার সফটওয়ার এবং তার মধ্যে চলতে শুরু করে আপনার পছন্দসই ওই গানটা। যদি একটা গ্রাফিক্স মানে ছবির ফাইল হয়, আপনি ক্লিক করা মাত্র খুলে যায় একটা ছবি দেখার বা ছবি আঁকার সফটওয়ার এবং তার মধ্যে ফুটে ওঠে আপনার নির্বাচিত ওই ফাইলের ভিতরকার ছবিটা। কিন্তু এটা ঘটছে কী করে?

আপনার ওই ক্লিক করার মুহূর্তটার মধ্যেও অনেক অনুমুহূর্ত থাকে। অপারেটিং সিস্টেমের কাছে আপনি প্রায় ব্রহ্মা। ব্রহ্মার এক আঁখিপল্লবপাতে যেমন ধরাধামে অগণিত যুগ বয়ে যায়, তেমনি আপনার এক মুহূর্তে অপারেটিং সিস্টেম তথা সিপিইউ-র কাছে কেটে যায় কোটি কোটি যুগ, যার মধ্যে অপারেটিং সিস্টেম পরপর করে চলে —

এ-ড্রাইভ বা '/mnt/floppy' নামে আপনি ঠিক কোন জায়গাটাকে বোঝাচ্ছেন সেটা দেখে নেওয়া

আপনি যে নামে আপনার ফাইলটাকে চিহ্নিত করছেন, সেই নামের ফাইলের জন্যে সিস্টেম কোন আইনোড বরাদ্দ রেখেছে সেটা দেখে নেওয়া, এর জন্যে একটা তালিকা আগেই করে রাখা আছে আইনোড টেবল নামে, আপনার পছন্দ করা জায়গাটার সমস্ত ফাইলকে মিলিয়ে (এই যে আইনোড তালিকার কথা বললাম, আইনোড হল একধরনের একটা সংখ্যা-সূচক ব্যবস্থা যা দিয়ে অপারেটিং সিস্টেম একটা ফাইলকে চেনে, আমরা এর সঙ্গে আমরা বিশদ ভাবে পরিচিত হব ছয় এবং সাত নম্বর দিনে গিয়ে)

সেই আইনোড তালিকা থেকেই পড়ে নেওয়া, আপনার নামের ফাইলটার পিছনে আসলে বাস্তব কোন কোন বাইট রয়েছে, তারপর সেই বাইটগুলোয় লিখে রাখা মোট তথ্যটাকে পড়ে নেওয়া

এবার সিস্টেমের ভিতরে আগে থেকেই বানিয়ে রাখা তালিকা থেকে দেখে নেওয়া আপনার দেওয়া ফাইলটা যে ধরনের, সেই ধরনের ফাইল খোলার জন্যে কোন প্রায়োগিক বা অ্যাপ্লিকেশন সফটওয়ার তার চালু করার কথা

এবার আপনার পছন্দ করা জায়গার মানে ডিরেক্টরির পছন্দ করা ফাইলটা থেকে পড়ে নেওয়া গোটা তথ্যটাকে, ওই ধরনের ফাইল খোলার সফটওয়ার চালিয়ে, তার মধ্যে হাজির করা।

জাস্ট ভাবুন আপনার একটিমাত্র নিদেন দুইমাত্র মাউসহেলনের অন্তরালে আপনার অপারেটিং সিস্টেমের মোট কতটা অঘোষিত নির্বাক শ্রম রয়ে যাচ্ছে। এবার, ধরুন এখানে গ্নু-লিনাক্স বা উইন্ডোজ নামের এই অপারেটিং সিস্টেমের নির্বাক মধ্যস্থতাটা নেই, একদম প্রথম দিকের কম্পিউটারে যা হত। সুইচ অন করা থেকেই মেশিনটা সম্পূর্ণ আপনার এবং একমাত্র আপনারই আজ্ঞারই অধীন — আপনিই রাখবেন বা মারবেন, এবং তার চেয়েও ডেনজারাস কথা এই যে কী করে রাখবেন বা কী করে মারবেন তার গোটাটাই আপনাকে জানতে হবে। এই রকম অবস্থায় আপনাকে বলতে হত — ফ্লপি-ড্রাইভের মধ্যের চাকাটা এত ডিগ্রিতে ঘোরাও, এতটা গতিতে, সেখানে প্লাস্টিকের পাতের গায়ে বিদ্যুৎ-আখরে উৎকীর্ণ সংখ্যামালা পড়ো, এই করতে গিয়ে ফ্লপি ড্রাইভের মুণ্ডটা এতটা বেঁকাও ... ইত্যাদি ইত্যাদি।

হার্ডওয়ারের এই জটিলতাগুলোর উপরে একটা আংশিক এবং সাঙ্কেতিক প্রলেপ তৈরি করে দিচ্ছে অপারেটিং সিস্টেম। আংশিক কারণ আপনার সামনে ওই সি-ড্রাইভ, এ-ড্রাইভ, বা '/user/share', '/mnt/floppy' ইত্যাদি সাঙ্কেতিক নামের মাধ্যমে সে তৈরি করে দিচ্ছে একটা ভারচুয়াল বা সাঙ্কেতিক জগত। আপনি ক্লিক করা মাত্র বা '`dir`' বা '`ls`' লিখে এন্টার মারা মাত্র আদত বাস্তব দুনিয়ায় সে কিছু একটা বাস্তব ঘটনা ঘটাচ্ছে যা আপনার কাছে পৌঁছচ্ছে ফাইল-তালিকা হাজির হওয়া নামক সাঙ্কেতিক ঘটনায়।

একটা জিনিষ খেয়াল করে রাখুন, এটা সারা পাঠমালাটা জুড়েই আমরা অনুসরণ করে চলব। যখনই কোনো ইংরিজি বর্ণমালায় লেখা নাম বা শব্দ উল্লেখ করছি আমরা, তার ফন্ট ব্যবহার করছি টাইমস নিউ রোমান, ধরুন '/user/share' বা '/mnt/floppy'। যখন কোনো আদেশ বা কমান্ড, মানে ঠিক যা টাইপ করে কমান্ড প্রম্পটে দিতে হয়, উল্লেখ করছি, যেমন '`ls /user/share`' বা '`mount /mnt/floppy`', তখন ফন্ট ব্যবহার করছি কুরিয়ের নিউ, মানে ফিক্সড উইডথ ফন্ট, যেখানে প্রতিটি অক্ষরই একই পরিমান চওড়া জায়গা নেয়, টাইপরাইটারে যেমন দেখি আমরা। যখন ওই কমান্ডটাকেও আবার নাম হিশেবে আনছি, তখন কিন্তু লিখছি টাইমস নিউ রোমানে। এর সঙ্গে আরো একটা জিনিষ করছি আমরা বাংলার বর্ণমালার সঙ্গে আয়তনগত সাযুয্য রাখার জন্যে টাইমস নিউ রোমান বা কুরিয়ের নিউকে তিন পয়েন্ট কমিয়ে দিচ্ছি। যখন বাংলা ফন্ট ১৪ পয়েন্ট, তখন ইংরিজি ১১, যখন বাংলা ১৩, ইংরিজি ১০, এই রকম।

একটা জিনিষ খুব ভালো করে মাথায় রাখুন, সিস্টেমকে বুঝতে এই ব্যাপারটা বারবার কাজে লাগবে — আপনার এবং অপারেটিং সিস্টেমের এই নাম দেওয়া এবং ব্যবহারের এই গোটাটাই একটা সাঙ্কেতিক ঘটনা। কারণ, আপনার কাছে যা আসছে অক্ষরের নিয়মে সাজানো পরপর বোধগম্য নাম হিসেবে, ধরুন, '/mnt/floppy/song.mp3' যে নাম আপনিই দিয়েছেন, ভৌত অর্থে কিন্তু এই গোটা নামটাই অর্থহীন, হার্ডডিস্কের গায়ে বাইটগুলোর ভৌত শরীরে এরকম কোনো নামই নেই, না 'mnt' বা 'floppy' নামের কোনো ডিরেক্টরি এবং সাবডিরেক্টরি, না 'song.mp3'

নামের কোনো ফাইল। '/mnt/floppy/song.mp3' নামের পিছনে অপারেটিং সিস্টেম যা খুঁজে নিচ্ছে এবং পড়ে তুলে নিচ্ছে নিজের মেমরিতে, যাতে এখুনি কোনো গান চালানোর অ্যাপ্লিকেশন দিয়ে তাকে খুলতে পারে, তারা আসলে কোটি কোটি অর্বুদ অর্বুদ ০ এবং ১ এর শেষহীন সারি। আদত হার্ডডিস্কের শরীরে ওই '/mnt/floppy/song.mp3' নামের কোনো অক্ষরের কোনো অর্থই নেই। কম্পিউটারের কাজের স্বার্থে অপারেটিং সিস্টেম ওই ০ আর ১-এর শেষহীন নিরবচ্ছিন্ন মিছিল থেকে এক একটা ০ আর ১-এর সমাহারের এক একটা সূচক তৈরি করেছে। ফাইল বানানোর ক্রিয়ায় যে সমাহারটা আপনিও বানিয়ে থাকতে পারেন, সিস্টেমও বানিয়ে থাকতে পারে। এই সমাহারগুলোর এক একটা নাম আছে — নামের তালিকা তৈরি হয়েছে, কিছু নাম আপনি দিয়েছেন, কিছু নাম সিস্টেম দিয়েছে, তাও আপনার বোঝার স্বার্থে। এরা হল ফাইল, যাদের এখন থেকে আপনি চিনবেন ওই নাম দিয়ে আর সিস্টেম চিনবে আইনোড তালিকার ওই সূচক দিয়ে।

এবার যখনি আপনি নাম-সর্বস্ব এই সাঙ্কেতিক দুনিয়ায় কোনো একটা নামের উপর কার্সর এনে ইঁদুরের পেট টিপলেন, বা নামটা টাইপ করে কোনো কমান্ড দিলেন, সে ওই সূচক তালিকা থেকে পড়ে নিল — আইনোড — আপনি যা করতে চাইছেন সেটা করল। আপনি ভুলে থাকতে পারলেন ওই সূচক, সূচকের পিছনে সেক্টরের শেষহীন সারি, তাদের মধ্যেকার চৌম্বক বলরেখার সাযুয্য যাকে সিস্টেম ০/১ হয়ে বিট বাইট হয়ে বর্ণমালা বলে চিনছে, ভুলে যেতে পারলেন একটা সেক্টর থেকে অন্য সেক্টরে যেতে আপনার হেডকে, সরি, আপনার না, আপনার সিস্টেমের হার্ড বা ফ্লপি ড্রাইভের হেডকে, কতটা নড়তে হবে, ভুলে যেতে পারলেন অ্যাকচুয়েটর আর মোটর আর তাদের সার্কিট আর তাদের বিদ্যুৎ-বিভবের ওঠাপড়া। আপনি শুধু নাম ধরে পুকারলেন, সে মাখোমাখো হয়ে গেল। কী দুর্ভাগা সেই মহাপুরুষ — শুধু একটা হার্ডডিস্কের বৃহদন্ত্র কখনো ঘাঁটতে হয়নি বলেই বলতে পেরেছিলেন — নামে কী এসে যায়।

শুধু হার্ড ফ্লপি বা কম্পাক্ট ডিস্ক কেন, আপনার প্রিন্টার হলে? তার কাঁটার পিছনে দোয়াতের উদরে কালির হিশেব, আমার প্রেতচুম্বিত ডটম্যাট্রিক্সে, বা আপনার জলচৌকি-পিচকিরি প্রিন্টারের মানে ডেস্কজেটের পিচকিরির — এই গোটাটাই আপনি ভুলে থাকছেন। আপনি 'প্রিন্ট করো' বলেই খালাস, মানে কমান্ড প্রম্পটে কমান্ড দিয়ে, বা, প্রিন্টারের আইকনের গায়ে ক্লিক করে। ওই মুহূর্তটাকেই ভাবুন — আপনি যখন প্রিন্টারের আইকনে ক্লিক করছেন, স্ক্রিনে আরো কিছু ঘটতে শুরু করছে, প্রিন্টের আইকনের গা থেকে একটা করে পাতার ছবি বেরোচ্ছে আর পরমাত্মায় লীন হয়ে যাচ্ছে, মানে প্রিন্ট হচ্ছে — ওই মুহূর্তটাতেই অপারেটিং সিস্টেম আরো কত কী করে চলেছে, আপনার দৃষ্টির অন্তরালে, ভেবে দেখেছেন কখনো? এটা ভাবতে যে হয়না সেটাই অপারেটিং সিস্টেমের সাফল্য।

আপনি মাউসের পেট টিপলেন, মাউসের নাড়িভুড়ির হিশেব রাখতে হল কিন্তু অপারেটিং সিস্টেমকে, স্ক্রিনে ফুটে ওঠা ডিটেইলস-এর সঙ্গে মিলিয়ে সেই মাউস-ইভেন্টকে বুঝল সে। মানে, ভিডিও মনিটরের সমস্ত খুঁটিনাটিও মাথায় রাখতে হয়েছে তাকে। সেই খুঁটিনাটির সঙ্গে মিলিয়ে সে মনিটরে করা আপনার মাউস-ইভেন্ট থেকে মিলিয়ে নিল, কোন ফাইলটাকে চাইছেন আপনি, আপনার ফাইলটা তুলল মেমরির যথাস্থানে। তার জন্যে তাকে মাথায় রাখতে হল মেমরির গঠনের খুঁটিনাটি। এবার মেমরি থেকে প্রিন্টার পোর্টে তাকে পাঠাতে হবে ওই তথ্যটা। তার মানে, কতটা হারে তাকে পাঠাতে হবে আপনার সিস্টেমের বাস বেয়ে সেই হিশেব করল, তার ডিটেইলস, প্রিন্টারের ডিটেইলস, প্রিন্টার পোর্টের ডিটেইলস, তবে পাঠাও তথ্যটা, একগুচ্ছ। আবার পড়ো, আবার স্ক্রিনের ছবি বদলাও, মুহূর্তে মুহূর্তে। অপারেটিং সিস্টেমের মুহূর্ত ঘটে চলেছে আপনার প্রতি মুহূর্তে মাত্র কয়েক কোটি হারে, গোটা সময়টা জুড়েই এই গোটা ঘাপলাটা হ্যান্ডল করছে একা একলা অপারেটিং সিস্টেম। হার্ডওয়ারের ওই খিঁটকেল দুনিয়ার সাপেক্ষে কম্পিউটার জগতের আর সব কিছু — সমস্ত কিছুকে বয়ে নিয়ে চলেছে অপারেটিং সিস্টেম। এর আরো কিছু খুঁটিনাটিতে আমরা আসব পরের দিন, দু নম্বর দিন, তারপরে আসতেই থাকব।

৪।। সিস্টেম প্রোগ্রাম এবং কমান্ড ইন্টারপ্রিটার

আমাদের তালিকাটায় দেখুন, মূল প্রোগ্রাম কাঠামো বা সিস্টেম প্রোগ্রামের এলাকা শুরু হয় অপারেটিং সিস্টেম থেকে। একদম নিচে হার্ডওয়ার থেকে গুনলে এটা ৪ নম্বর স্তর। এর উপরে মানে সিস্টেম প্রোগ্রামিং-এ এর উপরের মানে ৫ নম্বর স্তরে আছে তিনটে উপাদান — কম্পাইলার, এডিটর, কমান্ড ইন্টারপ্রিটার। অপারেটিং সিস্টেম এবং

তার ঠিক উপরের স্তরেই এই তিনটেকে মিলিয়ে তৈরি সিস্টেম প্রোগ্রামের এলাকা। কমান্ড ইন্টারপ্রিটার বলতে সেই সফটওয়ার যা আমাদের কমান্ডকে ইন্টারপ্রিট করে মেশিনের কাছে পৌঁছে দেয় এবং তাকে ওই কমান্ড পালন করায়। এই সংজ্ঞাটা আমায় শ্রীপঞ্চমীর পুণ্য প্রভাতে প্রণামনীয় দ্বিতীয় ঠাকুরের একটা গপ্পো মনে পড়াল। দর্শনের এমএ ক্লাসে এক মাস্টারমশাই, সে ব্যাটা ইংরেজ, নইলে এতাবৎ কলোনাইজার কনফিডেন্স আসবে কোত্থেকে, 'বিড়াল' নামক প্রাণীর পরিচয় দিতে গিয়ে বলে দিচ্ছে, 'ক্যাট' মানে হল 'এ ফেলাইন কোয়াড্রাপেড'। তবু যে ব্যাটা সাপ্লিমেন্ট অফ কপুলা দিয়ে বলেনি এটা আমাদের কলেনাইজডদের বাপের ভাগ্য। ওই সাহেবসুবোর স্টেটমেন্টটা খেয়াল করুন, আমাদের বিদ্যাচর্চার বিপুল সংখ্যক সংজ্ঞা ঠিক ওই বর্গের। 'মানে' ইত্যাদি দ্বিতীয় স্টেটমেন্টটাই যেখানে মূল মানেটাকে আরো ঘেঁটে দিচ্ছে, সংজ্ঞাটা শুনে যেখানে গুলিয়ে যাচ্ছে আরো, বিড়াল যে কাকে বলে সেটাই আপনি ভুলে যাচ্ছেন। কমান্ড ইন্টারপ্রিটারের সংজ্ঞাটা শুনে ঠিক তাই হল না? এবার একটা উদাহরণ দিয়ে বোঝা যাক।

ধরুন আমরা আগে যে উদাহরণটা দিলাম, '`ls /mnt/floppy`' বা '`dir a:`' — এখানে 'ls' বা 'dir' বা বলায় কম্পিউটারের অপারেটিং সিস্টেমকে যে ফ্লপি ড্রাইভে ঢোকানো ফ্লপিডিস্কের শরীরে লেখা ফাইল ডাইরেক্টরির তালিকাটাই হাজির করতে হবে স্ক্রিনে এটা অপারেটিং সিস্টেমকে বুঝিয়ে দিচ্ছে এই কমান্ড ইন্টারপ্রিটার। আসলে এখানে জটিল একটা প্রক্রিয়া শুরু হয় যা আমরা ঠিক করে বুঝতে শুরু করব পাঁচ নম্বর দিন থেকে। মোদ্দা ব্যাপারটা অনেকটা এইরকম, ব্যবহারকারীর সঙ্গে সরাসরি চারচক্ষুর মিলন হচ্ছে এই কমান্ড ইন্টারপ্রিটার-এর। কমান্ড ইন্টারপ্রিটার এই টাইপ করে দেওয়া পুরো জিনিষটা মানে '`ls /mnt/floppy`' শব্দগুলো থেকে 'ls' অংশটাকে আলাদা করে ফেলে। এই 'ls' টুকু হল আদত কমান্ড। অন্যটুকু মানে '/mnt/floppy' হল তার আর্গুমেন্ট বা যার উপর সে কমান্ডটা চালাবে। এর সঙ্গে আরো একটা অংশ থাকতে পারত, যার নাম 'অপশান' বা পছন্দ। ভালো করে আলোচনাটা ধরব আমরা ছয় নম্বর দিনের গোড়ায় গিয়ে।

ধরুন কমান্ডটা যদি হত '`ls -al /mnt/floppy`', তখন ওই হাইফেন বা '-'-এর পরের '`al`' অংশটা হত অপশান বা পছন্দ। এতে জানানো হত যে, ফ্লপিতে যদি এমন কোনো ফাইল থাকে যা সিস্টেম স্বাভাবিকভাবে দেখায় না, যেমন সিস্টেম ফাইল, সেটাকেও দেখাও। এবং দেখাও তাদের সম্পর্কিত সব খুঁটিনাটি, কে সেই ফাইল-এর মালিক, কে কে ফাইলটা পড়তে বা লিখতে বা চালাতে পারে, কখন ফাইলটা বানানো হয়েছিল, ফাইলের সাইজ, ইত্যাদি সমস্ত খুঁটিনাটি। এখানে 'al'-এর 'a' মানে অল, বোঝাচ্ছে সব ফাইল দেখাও, আর 'l' মানে লং, দীর্ঘ বিবরণের আকারে সমস্ত খুঁটিনাটি দেখাও ফাইলের। উইনডোজ প্রতিবেশে কমান্ডটা হত '`dir/a a:`' — দীর্ঘ বিবরণ দিতে বলার সুযোগটা উইনডোজে পাওয়ার ব্যবস্থা থাকেনা। যে অধিকার এবং অনুমতির নিরাপত্তাব্যবস্থার খুঁটিনাটি হিশেবে ওই উপাদানগুলো থাকে বা আসে, ফাইলসিস্টেমের গঠনের মধ্যে গেঁথে দেওয়া সেই নিরাপত্তাব্যবস্থাটাই নেই উইনডোজে। সাত নম্বর দিনে এই বিষয়ে আসব আমরা, কেন গ্নু-লিনাক্স একটা কাঠামোর ভাইরাস আক্রান্ত হওয়াটা সংজ্ঞাগত ভাবেই অসম্ভব। যাইহোক, ফিরে আসা যাক কমান্ড ইন্টারপ্রিটার-এর কথায়। অপশন সহ এই গোটা কমান্ডটা পালন করবে শেষ অব্দি অপারেটিং সিস্টেম, কিন্তু তার আগে এই গোটা কমান্ডটা কোথায় আছে, তার অপশানগুলোর ঠিক কেজো অর্থটা ঠিক কী দাঁড়াচ্ছে — এই গোটাটা আপনার কমান্ড পাওয়ার পর অপারেটিং সিস্টেমকে বুঝিয়ে দেওয়ার দায়িত্ব কমান্ড ইন্টারপ্রিটারের। পাঁচ নম্বর দিনে কিছুটা এবং তার পরে দশ নম্বর দিনের গোটাটা জুড়েই এই কমান্ড ইন্টারপ্রিটারের ব্যাকরণ শিখব আমরা। ভেবে দেখুন, আপনার কমান্ড হল এমন একটা কিছু যা কাজ করে, মানে, চলে। অর্থাৎ প্রোগ্রাম। মানে এমন একটা ফাইল যা চলে, আগেই যা বললাম। তার মানে এই ফাইলটা এমন কোথাও থাকতে হবে যেখান থেকে তাকে খুঁজে পেতে পারে কমান্ড ইন্টারপ্রিটার, এবং সেই কর্তব্যকর্মটা, তার সমস্ত শর্ত এবং বাধ্যতাসহ, তখন অপারেটিং সিস্টেমের হাতে হস্তান্তর করতে পারে। কোথায় কোথায় ফাইলটাকে খুঁজে দেখবে কমান্ড ইন্টারপ্রিটার বা শেল, গ্নু-লিনাক্সে সবচেয়ে জনপ্রিয় শেল ব্যাশ, সেই হদিশটা থাকে পথনির্দেশ বা 'PATH' নামক সিস্টেম ভ্যারিয়েবলে। এর কথাও পরে জানব আমরা।

মানে, মোদ্দা কথা হল, কমান্ড দিলে সেই কমান্ডটা তো একটা শব্দ এবং অক্ষরের সমাহার — সেই সমাহারের অর্থোদ্ধার করার দায়িত্ব কমান্ড ইন্টারপ্রিটার বা শেলের। ইনফ্যাক্ট, এবার অর্থ মোতাবেক কাজ করার দায়িত্ব অপারেটিং সিস্টেমকে ন্যস্ত করে কমান্ড ইন্টারপ্রিটার এবার ঘুমোতে চলে যায়। ছবি টবি দিয়ে এর একটা সুন্দর আলোচনা আছে পাঁচ নম্বর দিনে। তবে, সবসময় এই শেলকে খুব আলাদা করে খেয়াল করা যায়না। বিশেষ করে

যখন গুই বা গ্রাফিকাল ইউজার ইন্টারফেস-এ কাজ করছি, উইনডোজ যেমন, বা গ্নু-লিনাক্স-এ কেডিই বা গুহনোম বা ফ্লাক্সবক্স বা ব্লাকবক্স বা এনলাইটেনমেন্ট বা এক্সএফসিই ইত্যাদি, সেখানে কিন্তু সামনা সামনি প্রকাশ্যে এই কমান্ড ইন্টারপ্রিটারটাকে প্রায়ই ধরা যায়না। বোঝা যায়, ব্যাশে নেমে কী কী প্রসেস বা প্রক্রিয়া তখন সিস্টেমে চালু রয়েছে সেটা দেখতে চাইলে, 'ps aux' জাতীয় কমান্ড দিয়ে। দেখা যায় চলতে থাকা সবকটা প্রোগ্রামেরই উল্লেখ আছে তাতে। এবং তাদের সবারই মা-প্রক্রিয়া হল ব্যাশ, সেটাও মিলিয়ে নেওয়া, কী করে — সেটা আমরা পরে দেখব। এই প্রক্রিয়া বা প্রসেস মানে একটা প্রোগ্রামের একবার চলা, এর বাস্তব অর্থটা কী দাঁড়ায় সেটা আমরা ধীরে ধীরে বুঝব।

ধরুন 'move' বা 'mv' বা কমান্ডটাকেই, উইন্ডোজ-এর বেলায় 'move', আর লিনাক্স-এর বেলায় 'mv'। উইন্ডোজ-এ যদি আমরা মাউস ক্লিক করে একটা ফাইলকে অন্য জায়গায় নড়াই সেখানেও সামনের ওই ফোল্ডার থেকে ফোল্ডারে ফাইল উড়ে যাওয়ার ছবির পিছনে আসলে কাজ করে এই কমান্ডটাই, আমাদের অগোচরে। আর প্রত্যক্ষে এটাকে আমরা দেখতে পাই যখন এমএস-ডস প্রম্পট খুলে সরাসরি কমান্ড ইন্টারপ্রিটার দিয়ে কাজ করি। আমরা মাউস টিপে বা কিবোর্ড থেকে ওই কমান্ড-টা দিলে কী করতে হবে সেটা মেশিনকে জানায় কমান্ড ইন্টারপ্রিটার। যেমন উইন্ডোজ-এর বেলায় এই কমান্ড ইন্টারপ্রিটারের নাম 'command.com'। এই ফাইলটা থাকে উইন্ডোজ-এর রুট ডিরেক্টরিতে, মানে স্বাভাবিক একটা সিস্টেমের বেলায় C:\ -এ। শুধু এই ফাইলটা নয় অবশ্য, কমান্ড ইন্টারপ্রিটার-এর গোটা আবহটা উইনডোজ প্রতিবেশে বানিয়ে তোলে আরো কিছু ফাইল, যেমন 'msdos.sys' আর 'io.sys'। এরাও থাকে ওই রুট ডিরেক্টরিতেই। তবে পরের দুটো গোপন ফাইল, দেখতে চাইলে কমান্ড প্রম্পটে দিতে হয় 'dir/a c:' বা গুই মোডে থাকলে এক্সপ্লোরারের ভিউ মেনুতে 'শো অল ফাইলস' অপশান অন করতে হয়। আর লিনাক্স-এর বেলায় এই কাজটা করে শেল, মানে সচরাচর যার নাম bash — যদিও এছাড়াও অন্য শেল আছে। তিনি থাকেন রুট ডিরেক্টরির ভিতর '/bin' ডিরেক্টরিতে, তার ঠিকানাটা হল '/bin/bash'। এই ঠিকানার ব্যাপারটাও ঠিক ভাবে বুঝতে আমাদের ছয় নম্বর দিন অব্দি ওয়েট করতে হবে।

৫।। এডিটর

দ্বিতীয়টা হল এডিটর, যাকে বলা যায় সবচেয়ে সরল সাদামাঠা ওয়ার্ড প্রসেসর। দিন শূন্যয় আমরা যাকে বিশুদ্ধ টেক্সট-এর বিশুদ্ধ এডিটর বলে ডেকেছিলাম। ওয়ার্ড প্রসেসর বলতে আমাদের মাথায় থাকে ওপেন সোর্স জগতের ওপেন অফিস অর্গ রাইটার বা মাইক্রোসফট-এর ওয়ার্ড বা লোটাস-এর স্মার্ট স্যুট বা সিমানটেক-এর পেজমেকার। কিন্তু ভিতরে ভিতরে এর প্রত্যেকটাই হল একটা সরল সাদামাঠা এডিটর যা অক্ষরের সমাহারকে হাজির করে, এবং সেই সরল সাদামাটা এডিটরের সঙ্গে বাড়তি আরো কিছু ফর্মাটিং — অক্ষরগুলোর কোনো বিশেষ আয়তন বা রং বা বিশেষ কোনো চেহারা বা ফন্ট বা অন্য কিছু যেমন ব্লক বা ইটালিক বা আন্ডারলাইন ইত্যাদি আছে কিনা, প্যারাগ্রাফগুলোর কোনো ফর্মাটিং আছে কিনা, , বা কোনো সারণী বা ছবি বা কোনো বিশেষ কিছু আছে কিনা, ইত্যাদি। তাদের ঠিক কী ভাবে সাজানো আছে পাতায়, মানে তাদের লে-আউট ইত্যাদি।

এই গোটা গল্পটাকে বুঝতে পারবেন বেশ ভালো করে, হাতেকলমে, যদি একটা ওয়ার্ডপ্রসেসর ফাইলকে কমান্ড এডিটর দিয়ে খোলেন। কমান্ড এডিটর নামটা কেন তাতো আগেই বলেছি, এক বা একাধিক কমান্ডের তালিকাকে লিখে ফেলা যায় এবং এডিট করা যায় এই কমান্ড এডিটর দিয়ে, কোনো ধরনের কোনো ফর্মাটিং-এর ধার ধারেনা, বিশুদ্ধ নিখাদ আছোলা টেক্সট। ঠিক যেমন কম্পিউটারের অপারেটিং সিস্টেম যেমন কোনো কমান্ড পালন করতে গিয়ে বাড়তি কোনো ফর্মাটিং পায় তার ভিতর, মাঝখান থেকে কমান্ড বা আদেশটাই ধরতে পারবে না, এবং তাই পালন করতে পারবে না। এই পরপর কমান্ডগুলোকে একসঙ্গে এনে যে ফাইলগুলো আমরা তৈরি করি কমান্ড এডিটর দিয়ে, তাদের চালু নাম স্ক্রিপ্ট । এই পাঠমালার শেষ মানে দশ নম্বর দিনটা আমরা গোটাটাই প্রায় ব্যয় করব ব্যাশ শেলকে কাজ করানোর নানা ধরনের এই স্ক্রিপ্ট দিয়ে। এই স্ক্রিপ্টগুলো গ্নু-লিনাক্স মেশিনে ভারি গুরুত্বপূর্ণ একটা ব্যাপার। এর মধ্যে থাকে পরপর যে কমান্ডগুলো আপনি কম্পিউটারকে দিয়ে করাতে চান। কম্পিউটার যে কমান্ড পাবে সেখানে তো ফর্মাটিং থাকলে চলবে না। ফর্মাটিং মানে তো মূল বিশুদ্ধ টেক্সট যোগ আরো কিছু চিহ্ন। যে চিহ্নগুলো ওই ফর্মাটিংকে হাজির করবেন। যে কোনো ওয়ার্ডপ্রসেসরের মধ্যে 'শো ননপ্রিন্টিং ক্যারেকটারস'

অপশানটা অন করে দিলে পাতা জুড়ে লেখার সঙ্গে সঙ্গে যে চিত্তির-বিচিত্তিরগুলো ফুটে ওঠে। সেই চিহ্নগুলো মূল কমান্ডকেই আটকে দেবে, কম্পিউটারের কাছে সঠিক কমান্ড পৌঁছেই দেওয়া যাবেনা।

এটা একদম হাড়ে হাড়ে বুঝবেন, ধরুন আপনি ওয়ার্ডপ্রসেসর ওপন-অফিসে লেখা একটা ফাইল যদি কমান্ড এডিটর ইম্যাক্স বা ভিম দিয়ে খোলেন, বা এমএসওয়ার্ডে লেখা ফাইল যদি নোটপ্যাড দিয়ে খোলেন। মূল টেক্সট-এর একটা জায়গা চিনে রাখুন — তারপর সার্চ দিন সেই অক্ষরমালা দিয়ে, টেকনিকাল ভাষায় যার নাম স্ট্রিং, দেখবেন, সবটাই আছে, পুরো টেক্সটাই, শুধু কিছু অক্ষর কিছু চিহ্ন একটু অন্য চেহারায় দেখতে পারেন, যাদের যে চেহারায় দেখানোর উপায় নেই ওই সাদামাঠা কমান্ড এডিটরের। এবং পাবেন আরো বহু চিহ্ন এবং সংখ্যা, প্রচুর, যাদের কোনো অর্থ করতে পারবেন না।

এই সরল আকারে এডিটরকে পেতে লিনাক্স জগতে আছে প্রচুর কিছু, যার মধ্যে প্রমুখতম হল ইম্যাক্স এবং ভিম। এবং উইন্ডোজ জগতে আছে এডিট এবং নোটপ্যাড। এর মধ্যে এডিট বস্তুটা ডস-এর সময় থেকেই আছে — এখনো ২০০০ এবং এক্সপিতেও দেওয়া থাকে। ডস প্রম্পট-এ সরাসরি ‘`edit textfile`’ লিখে এন্টার মারলে দেখবেন একটা আদিম কালের চাঁদিম হিম গোছের কালচে ধূসর উইন্ডো খুলে গেল, তার মধ্যে একটা ফাইল খোলা রয়েছে, যার নাম ‘textfile’ এবং সেখানে এবার আপনি যা টাইপ করবেন সেটাই ফুটে উঠবে এই ফাইলে। শেষে আপনি সেভ করে বেরোবেন।

এই সাদামাটা কমান্ড এডিটরগুলো প্রাথমিক ভাবে যা কাজ করে তা হল অনেক কমান্ডকে একসঙ্গে জুড়ে একত্রে হাজির করা। কোনো ফর্মাটিং কোনো আড়ম্বর কোনো কারুশিল্প ব্যতিরেকে। কারণ ওই কারুশিল্পের প্রয়োজন পড়ে আমাদের, কম্পিউটারের নয়। যেমন উইন্ডোজ-এর ব্যাচ ফাইল লেখা হয়। উইন্ডোজ ইনস্টলেশন-এ C:\ বা রুট-ডিরেক্টরিতে যেমন আমরা ব্যবহার করতাম, খুব কাজে লাগত ফাইলটা, ‘autoexec.bat’। এছাড়াও নানা কাজে ছোটখাটো নানা ব্যাচ ফাইল ব্যবহার করেছি। আমার মত বুড়ো কেউ যদি এই লেখাটা পড়ে, তারই বোধহয় এটা পরিচিত লাগতে পারে, ডস-প্রম্পটে ওয়ার্ডস্টার দিয়ে টাইপরাইটার হিশেবে কম্পিউটার ব্যবহার করা দিয়ে যাদের কম্পিউটারের সঙ্গে ফোরপ্লে শুরু হয়েছিল। এত আইডেন্টিফাই করেছিলাম রোমান পোলানস্কির বিটার মুন সিনেমার ওই বুড়ো লেখকের সঙ্গে — ব্যাটা ওয়ার্ডস্টারে লিখছে। তখন নানা কাজে লাগত ব্যাচ ফাইল।

এই ব্যাচ ফাইলের পরপর লেখা কমান্ডগুলোকে একের পর এক পালন করতে থাকে কম্পিউটার। একবার ভাবুন তো — কম্পিউটার প্রোগ্রাম ব্যাপারটা আসলে কী? পরপর করে যাওয়ার দীর্ঘ আদেশ তালিকা ছাড়া? সেদিক থেকে এই ব্যাচ ফাইলকে সি বা ফোর্ট্রান বা সিপ্লাসপ্লাস বা জাভা বা পার্ল বা পাইথন বা অন্য যে কোনো কম্পিউটার ল্যাংগুয়েজ-এ লেখা প্রোগ্রামের ডিম ভাবা যেতে পারে যা ফুটে এখনো ছানা বেরোয় নি। খুব বিশদ করে এই প্রসঙ্গটায় আমরা আসব পাঠমালার শেষ দিনে।

এবার ধরুন, এই কমান্ড এডিটর, বা নির্দেশ-সম্পাদক, যে নামেই আপনি ডাকুন, তাই দিয়ে আপনি একটা প্রোগ্রাম লিখলেন। আট নয় বা দশ নম্বর দিনে এরকম শেল স্ক্রিপ্ট আমাদের লাটকে লাট পড়তে হবে। তখন দেখবেন, যেটুকু কম্পিউটার ভাষা জানলে ওটা বোঝা যায় সেটুকু জানা থাকলে তো বুঝতেই পারবেন গোটাটা, আর না-থাকলেও অন্তত কিছু বুঝতে পারবেন। মানে সেটা একটা বোধ্য জগত। বিশেষ কিছু শব্দের বা ব্যবহারের বিশেষ অর্থগুলো জানা থাকলেই যার মানে বোঝা যায়। যেমন ধরুন একটা শক্ত এবং অপরিচিত বিষয়ের বই খুললে আমাদের হয়, বুঝতে পারি যে এটা বোঝার যোগ্য, শুধু বিষয়টা জানিনা বলে বুঝতে পারছি না। এর সঙ্গে কিন্তু একটা বিরাট ফারাক আছে আমাদের পরিচিত প্রোগ্রামগুলোর, যা দিয়ে আমরা লেখালেখির বা গানশোনার বা ছবি-আঁকার অ্যাপ্লিকেশন চালাই। ধরুন, গ্নু-লিনাক্স জগতের অ্যাবি-ওয়ার্ড আর এমএস-উইনডোজের এমএস-ওয়ার্ড, এরা দুজনেই দুটো প্রোগ্রাম, মানে, এমন ফাইল যে চলে। আমরা যখনি ওয়ার্ড-প্রসেসর চালাই, তখন আমাদের ক্লিক করা বা কিবোর্ডে কমান্ড দেওয়া মাত্র অপারেটিং সিস্টেম আসলে এই প্রোগ্রামগুলোকে চালায়। এই রকম কোনো ওয়ার্ডপ্রসেসর, বা গ্রাফিক্স প্রোগ্রাম, বা অন্য কোনো প্রোগ্রামকে সরল সাদামাটা কমান্ড এডিটর দিয়ে খুলুন তো? কী পেলেন, পড়ে কী বুঝছেন? কী দেখছেন? ঠিক আমাদের শূন্য নম্বর দিনের ভিশুয়াল চ্যাংড়ামোর মত একটা স্ক্রিন তো? তাহলে? এই তফাতটা কেন?

এটা বুঝতে গেলে আমাদের কম্পাইলার ব্যাপারটা কী সেটা জানতে হবে। কম্পাইলার, আগেই বলেছি, একটা দোভাষি, যা মানববোধ্য ভাষাকে কম্পিউটার বোধ্য করে তোলে। অনুবাদ করে যেমন আমরা অন্য ভাষার গল্প পড়ি। মানে বুঝতে পারি। এই কম্পাইলার হল আমাদের একটু আগে করা টেবিলে সিস্টেম প্রোগ্রাম স্তরে আমাদের তালিকার শেষ বস্তুটা।

৬।। কম্পাইলার

একটা খুব ছোট্ট সি প্রোগ্রাম ভাবা যাক। ধরুন তার নাম দিয়েছেন 'gadha.c' — একে বলে সি-এর কোড ফাইল। লেখা হয়েছে কোনো একটা কমান্ড এডিটর দিয়ে। এটা মোটামুটি সর্বজনগ্রাহ্য একটা প্রথা, সি নামের প্রোগ্রামিং ভাষায় লেখা কোড মানে প্রোগ্রাম ফাইলের নাম শেষ হবে, '.c' দিয়ে। সুগারকে শুগার বা লিজেন্ড-কে লেজেন্ড বা ডেভেলপমেন্ট-কে ডিভলেপমন্ট বলার মত, ধাপে ধাপে দামীতর বকলসের কুকুর হওয়ার প্রক্রিয়ায়, নিউব্যারাকপুর কিশলয় ইশকুলের 'টাইটেল' শব্দটা দিব্য বদলে গেছিল কলেজস্ট্রিটের 'সারনেম' বা 'পদবি' বা আরো পরে 'লাস্টনেম' শব্দে, রীতিমত লজ্জা পেতাম একসময়, ভুল বলে ফেললাম না তো, অ্যান্ড নাউ দ্যাট আই অ্যাম নাইন্টিথ্রি, ডোন্ট কেয়ার এ ড্যাম ইউ সি, তাই টাইটেলেই চলুক। ফাইলনামের এই '.c' টাইটেল দেখলেই আপনি বা আপনার সিস্টেম দুজনেই চিনে নিতে পারবেন এটায় ভরা আছে সি ভাষার কোড। এবার এই 'gadha.c' ফাইলের কোডটা দেখুন।

```
#include <stdio.h>
int main(void)
{
     printf("\nKire Gadha!!!\n");
     return 0;
}
```

এই প্রোগ্রামটা আর কিছুই না — কম্পিউটারকে বলে উপরে আর নিচে একটা করে ফাঁকা লাইন রেখে স্ক্রিনে ইংরিজি অক্ষরে ফুটিয়ে তোলো — 'কিরে গাধা!!!'। যার 'কিরে গাধা!!!' ফুটিয়ে তুলতে এতগুলো লাইন নির্দেশ লাগে, তাকে গাধা ছাড়া কী বলবেন? এই লাইনগুলোয় মানে 'gadha.c' প্রোগ্রামটায় কী আছে, তার একটা আলগা আন্দাজ নেওয়ার চেষ্টা করা যাক।

> উপরের দুটো লাইন, '#include <stdio.h>', 'int main(void)' এবং '{', আর একদম শেষের লাইন, '}' — এই চারটে লাইন অনেকটা সি-সঙ্গীতের ধ্রুবপদের মত, যা কেউ শোনেনা, গানের মাঝে ওইখানটায় নামিয়ে নেয় পেন্ডিং হাঁচিটা। এবং পাঁচ নম্বর লাইনও, মানে 'return 0;', আদতে সেই একই, যে ধ্রুবপদ দিয়েছো বাঁধি। কিন্তু মজার কথা কী বলুন তো, সেটা বোঝার মত বুদ্ধিটুকুও আপনার মেশিনের নেই, তাকে ওই লাইনগুলো দিয়ে দিতেই হবে। নইলে একটা দায়িত্বশীল নাগরিকের ভোটারলিস্টেই আপনি প্রোগ্রামটাকে তুলতে পারবেন না, ভোট দেওয়া তো দূরের কথা। এর উল্টোদিকে আবার, কীর্তনের মূল লাইনেরও কোনো অর্থ নেই মালের কাছে। এখানে 'কিরে গাধা' না বলে 'কিগো সোনা' বললেও তার জ্যাবড়া মুখের একটা রেখাও বদলাত না। দাঁড়ান দাঁড়ান, কম্পিউটারের মুখ কোনটা — এটা কোনোদিন ভাবিনি তো?
>
> দোহারকির প্রথম লাইন, মানে '#include <stdio.h>', জানায়, সি-এর নিজস্ব সঙ্কেতে, যে, এই প্রোগ্রামে আমরা সি-এর স্বাভাবিক ইনপুট আউটপুট ফাংশনকে ব্যবহার করব — স্ট্যান্ডার্ড ইনপুট আউটপুট — স্ট্যান্ডার্ড-এর এসটিডি (std) এবং ইনপুটআউটপুট-এর আইও (io)। আর ডটএইচ (.h) হল সেই সাফিক্স যা সি-র সমস্ত এই জাতীয় ফাইলের শেষে থাকে, হেডার ফাইল — যা থেকে এদের চেনা যায়। কেন এটাকে দোহারকি বলেছিলাম জানেন, ধরুন, গান গেয়ে ওঠার গায়ক যদি বলে, বন্ধুগণ, এবার আমি আবার গলা ব্যবহার করে আপনাদের গান শোনাতে যাচ্ছি। অ্যাজ-ইফ, খুব চেষ্টা করলে সে তার কান বা পা ব্যবহার করেও গাইতে পারত।
>
> দ্বিতীয় লাইনটা, মানে 'int main(void)', আমাদের জানায় যে, 'gadha.c' নামের এই কুচো প্রোগ্রামটায় আমরা যে ফাংশনটা গড়ে তুলব তা আমাদের শব্দ ফুটিয়ে তোলার কাজটা করে দেবে কিন্তু তার ফাংশন হিশেবে চরিত্রটা পূর্ণসংখ্যার মানের, মানে ইন্টিজার, যার চিহ্ন ইন্ট (int)' এবং এই প্রোগ্রামটার জন্যে তার

নিজের কোনো ভ্যালু বা মান থাকবে না। অর্থাৎ, সেটা শূন্যগর্ভ বা ভয়েড (void)। আর এই ফাংশনটা মূল বা মেইন (main) ফাংশন এই অর্থে যে তার সঙ্গে আরো চামচা ফাংশন থাকতেই পারত, অন্য অনেক প্রোগ্রামে তা থাকেও। এই কুচো প্রোগ্রামটার বেলায় যা নেই। এই ফাংশন ব্যাপারটা সি-ভাষার ব্যাকরণ মোতাবেক বাধ্যতামূলক, সেই জায়গাটুকু বানিয়ে দেয় — যেখানে দাঁড়িয়ে সি গোটা কাজটা করে। যাকগে, সেই আলোচনার জায়গা এটা নয়।

তিন এবং ছয় নম্বর লাইন মানে শুরু আর শেষের ব্রেস বা সেকেন্ড ব্রাকেট মানে, '{' আর '}' — এরা দুজনে মিলে দেখায় ওই মেইন ফাংশনের শুরু আর শেষ।

পাঁচ নম্বর লাইন, মানে 'return 0;', ওই শূন্যগর্ভ ফাংশনের শূন্যগর্ভতাকেই আর একবার ঘোষণা করে। জানায় যে, সত্যিই ফাংশনটা একটা শূন্য ভ্যালু ফিরিয়ে দিয়েছে। আসলে এটা একটা সিগনাল, ইউনিক্স এবং সি-তে প্রবল ভাবে ব্যবহৃত সিগনালটা, এর অর্থ এই যে, কাজটা সফল ভাবে শেষ হয়েছে। লাইনটার শেষে যে যতিচিহ্নটা, ';', এটা সি নামক কম্পিউটার ভাষার ব্যাকরণ অনুযায়ী কোনো লাইনের বাধ্যতামূলক যতিচিহ্ন। দেখুন দুটো সেকেন্ড ব্রাকেটের মধ্যে থাকা প্রোগ্রামের যে মূল অংশ, তার দুটো লাইনই শেষ হয়েছে এই যতিচিহ্নে।

একমাত্র পড়ে রইল 'gadha.c' প্রোগ্রামের চার নম্বর লাইন 'printf("\nKire Gadha!!!\n");'। আদতে প্রোগ্রাম মানে এটুকুই। এরও শেষে সেই বাধ্যতামূলক যতিচিহ্ন ';'। সেটাকে বাদ দিলে যেটুকু রইল তার মধ্যে একটা হল সি ভাষার একটা নির্দেশ, প্রিন্ট ফাংশন 'printf'। প্রিন্টএফ ফাংশন দিয়ে জানানো হচ্ছে, তার পরের ব্রাকেটের মধ্যে, দুই ডাবল-কোট বা '"' চিহ্নের মধ্যের গোটাটা স্ক্রিনে প্রিন্ট করতে মানে ফুটিয়ে তুলতে। মধ্যের অংশটা খেয়াল করুন, '\nKire Gadha!!!\n'। এই অংশটার শুরু আর শেষে রয়েছে দুই '\n', যার মানে, আমরা বোধহয় আন্দাজ করতে পারছি, নিউলাইন বা নতুন-লাইন। মানে দুবার জানানো হচ্ছে এক লাইন করে ছাড়তে, একবার ফুটিয়ে তোলা কথাটার শুরুতে, আর একবার শেষে। আর এই দুই নিউলাইনের মধ্যে রয়েছে ফুটিয়ে তোলার বাক্যাংশটা, আমাদের সেই সম্ভাষণ, 'Kire Gadha!!!'। এই দুর্ধর্ষ রকমের সমাদরপ্রবণ লাইনটাকে আপনি যাতে উপরে এবং নিচে দুটো ফাঁকা মানে কালো লাইনের প্রেক্ষাপটে তার সমূহ স্পষ্টতায় স্ক্রিনের কালো শরীরে ফুটে উঠতে দেখতে পারেন।

আপনারা যারা একটুও সি জানেন না তাদের কাছেও নিশ্চয়ই আর জটিল লাগছে না। ম্যাক্সিমাম একটু অপরিচিত লাগতে পারে গোটা ব্যাপারটা, কিন্তু মোদ্দা ব্যাপারটা খায় না মাথায় দেয় সেটা বেশ ভালোই বোঝা যাচ্ছে। কিন্তু মজার কথা কী বলুন তো? এই এটুকুও কিন্তু একটা কম্পিউটারের কাছে দুর্বোধ্য নয় শুধু, সম্পূর্ণ অবোধ্য। এই অবোধ্য ব্যাপারটাকে কম্পিউটারের কাছে বোধ্য তাই পালনীয় করে তোলে কম্পাইলার।

বলেছি আগেই, কম্পিউটার শুধু বোঝে ০ আর ১। একটা সার্কিটে হয় বিদ্যুৎ আছে, অথবা নেই। এবং এর সুবঘীর বোধ এবং জ্ঞানভাণ্ডারের এখানেই শেষ নয়। শুধু ০ আর ১ বোঝাই নয়, আমাদের কম্পু-বাবু এমনকি তাদের যোগ-ও করতে পারেন। ০ আর ০ মানে শূন্য, এবং ০ আর ১ মানে ১। কম্পিউটার যাই করুক, সার্চ ফর এক্সট্রাটেরেস্ট্রিয়াল ইন্টেলিজেন্স থেকে হিউম্যান জিনোম, সবই এই ০ আর ১ আর যোগ। এবার গুণ মানে অনেক যোগ, বিয়োগ মানে যোগের উল্টো। আর ভাগ মানে গুণের উল্টো। এভাবেই গোটাটা। তাই কম্পিউটার এমনই গাধা যে আমাদের ওই গাধোচিত প্রোগ্রামটিকেও সরাসরি দিলে কম্পিউটার কিছুই বুঝতে পারেনা।

সি বা যে কোনো কম্পিউটার ভাষাই, সেই অর্থে তাই, কম্পিউটারের ভাষা নয়, কম্পিউটার চালকের ভাষা, প্রোগ্রামচির ভাষা, কী করিতে হইবে এটা কম্পিউটারকে যে বলে দেয় সেই প্রোগ্রামের রচয়িতার ভাষা। কম্পিউটার কাজ করে নিতান্তই ওই একটু আগে দেখানো টেবিলের ১ আর ২ নম্বর স্তর মানে ভৌত উপাদান আর মাইক্রোআর্কিটেকচার দিয়ে। সেই জগতের উপর চলে ওই ০ আর ১ আর +, মানে ০ আর ১ দিয়ে লেখা তথ্য এবং তাদের যোগ করে চটকানো। সব জটিলতার তথ্যই আসলে তৈরি এই সরলতম ০ আর ১ দিয়ে। এবং সব জটিলতার সব চটকানোই তৈরি এই যোগক্রিয়ার নানা রকম সমাহার দিয়ে।

এবার, আপনি আপনার কম্পিউটারকে দিয়ে যাই করাতে চান না কেন, তার একটা নিজস্ব প্রক্রিয়া আছে। এই প্রক্রিয়ার প্রাথমিক ধাপ দুটো। এক, কী করাতে চান, সেটাকে যুক্তিশীল এবং বোধ্য রকমে পরপর লিখে ফেলা। এ

কাজেই আমাদের প্রয়োজন পড়ে কম্পিউটার ভাষা। সি যে ভাষাদের অন্যতম। এই সি ভাষা আমাদের কাছে বোধ্য, কারণ আমরা সেই ভাষা না-বুঝলে সেই ভাষায় নির্দেশ দেব কী করে? এবার ধাপ দুই, মানে, এই সি ভাষায় লেখা নির্দেশমালাকে কম্পিউটার-বোধ্য করে তোলা। যে কাজটা করে কম্পাইলার। সি থেকে সরাসরি মেশিন ল্যাংগুয়েজে নেমে প্রোগ্রামটার একটা মেশিন-ভাষায় মেশিন বোধ্য অনুবাদ তৈরি করাই কম্পাইলারের কাজ। এই প্রতিরূপটা আর আমাদের কাছে বোধ্য নয়, হওয়ার কোনো প্রয়োজনও নেই। তাই, একটা চালনীয় বা এক্সিকিউটেবল ফাইল এডিটর দিয়ে খুলে আমরা ওই সব বিচিত্র জিনিষ দেখতে পারি, ইটি এবং অ্যালিয়েন যে লিপিতে নিজেদের মধ্যে চিঠি-চালাচালি করে। ছিল একটা টেক্সট ফাইল, তাতে সি ভাষায় লেখা মানববোধ্য আদেশ। কম্পাইল করে সেটা থেকে পাচ্ছি একটা চালনীয় বা এক্সিকিউটেবল ফাইল। যাতে ভরা কম্পিউটারবোধ্য মেশিনভাষার আদেশমালা। সেই আদেশ কম্পিউটার বুঝতে পারে, তাই সেই ফাইল সরাসরি চালানো যায়, কমান্ড প্রম্পটে সেই ফাইলের নাম লিখে এন্টার মারা মানে কম্পিউটারকে জানানো, তুমি এই এক্সিকিউটেবল ফাইলে ভরা আদেশগুলো পালন করো, মানে, মোদ্দা কথাও তুমি এই ফাইলটা চালাও।

এই কম্পাইলার আছে হরেক ব্রান্ডের, নানা কোম্পানির। উইন্ডোজ জগতে মাইক্রোসফট বা বোরল্যান্ড ডাভিডি বা এখন ইনটেলের, এবং অন্য আরো অনেক আছে। এবং সবাই নিখাদ কম্পাইলারও নয়, তাতেও নানা ধাঁচের জটিলতা আছে। নানা বিশেষ বিশেষ কাজের বিশেষ বিশেষ ব্যবস্থা আছে। গ্নু-লিনাক্স জগতে আছে জিসিসি, যা একই সঙ্গে সি, সি প্লাস প্লাস, ফোর্ট্রান, অবজেক্টিভ সি, জাভা — যাবতীয় ভাষায় লেখা প্রোগ্রামই কম্পাইল করতে পারে। মানে আপনি যাকে কমান্ড বলে বুঝতে পারছেন কম্পিউটার ভাষা জানার দৌলতে তাকে কম্পিউটারের কাছেও বোধ্য করে তুলতে পারে, সেই কাজটা তাকে দিয়ে করিয়ে নেওয়ার ব্যবস্থা করতে পারে।

তাহলে, কম্পাইলার ঠিক কী করছে? কম্পাইল করা মানে কী? পরে আমরা এটায় বিশদ ভাবে আসব, আপাতত একটা আলগা আন্দাজ তৈরি করে রাখি। এই যে সি নামক কম্পিউটার ভাষায় লেখা আদেশে ভরা কোড ফাইলটা, মানে 'gadha.c', এটা তো বললাম, কম্পিউটারের কাছে অবোধ্য। কম্পাইলার এবার তাকে কম্পিউটার বোধ্য করে তুলবে, যাতে কম্পিউটার সেই আদেশগুলোকে বুঝতে পারে, তাই পালন করতে পারে। স্ক্রিনে ফুটিয়ে তুলতে পারে সেই চূড়ান্ত সমাদর, 'কিরে গাধা!!!'।

ধরুন এটা একটা গ্নু-লিনাক্স মেশিন, এবং আমাদের কম্পাইলার জিসিসি (gcc — **G**NU-**C**ompiler-**C**ollection)। এই অবস্থায়, 'gadha.c' ফাইলটা লেখা এবং সেভ করার পর, আমরা কমান্ড প্রম্পটে টাইপ করে জিসিসি-কে কমান্ড দেব, '`gcc -o gadha gadha.c`'। যার অর্থ, 'gcc' নামের কম্পাইলার ব্যবহার করে এই 'gadha.c' ফাইলটাকে কম্পাইল করো এবং কম্পাইল করে বানানো সেই চালনীয় বা এক্সিকিউটেবল ফাইলটার নাম দাও 'gadha'। এই 'gadha' হল সেই এক্সিকিউটেবল ফাইল বা প্রোগ্রাম। এবার কমান্ড প্রম্পটে গাধার নাম লিখে এন্টার মারলেই 'gadha' নামের প্রোগ্রামটা চলবে। মানে, 'gadha' নামের এক্সিকিউটেবল ফাইলে ভরা মেশিনবোধ্য আদেশগুলোকে পালন করবে। এই মেশিনবোধ্য আদেশগুলো এসেছিল 'gadha.c' নামক টেক্সটফাইলে ভরা মানববোধ্য আদেশগুলোকে জিসিসি নামের কম্পাইলার দিয়ে কম্পাইল করে।

এটা আমাদের অতিসরলীকৃত উদাহরণ। আদতে কমান্ডটা এত সহজ হয়না, সেখানে আরো অনেক কিছু থাকে। ফাইল লেখার সময় কোনো ভুল হলে সাবধান বাণী বা ওয়ার্নিং দিতে বলা থাকে কম্পাইলারকে। থাকে পোকা-ছাড়ানো বা ডিবাগিং সংক্রান্ত তথ্য ফাইলের মধ্যেই রেখে দেওয়ার আদেশ। থাকে অপ্টিমাইজেশন, মানে, কম্পিউটারের রসদের সঙ্গে নিজেকে মানিয়ে নেওয়ার আদেশ। ইত্যাদি। আবার, শুধু গ্নু-লিনাক্সে কেন, 'gadha.c' নামের কোড ফাইল কম্পাইল করা যাবে উইনডোজ প্লাটফর্মের কোনো সি-কম্পাইলার দিয়েও, সি নামক ভাষাটা তো একই। কিন্তু সেই কম্পাইল করার কমান্ডটা আলাদা হবে। এই একই কমান্ডে জিসিসি দিয়েই উইনডোজ প্ল্যাটফর্মেই কম্পাইল করা যায় — ডিজেজিপিপি নাম কম্পাইলারটার, ইম্যাক্স দিয়ে কোড লেখো এবং জিসিসি দিয়ে সি প্রোগ্রাম করো উইনডোজ-এর মত খাজা প্লাটফর্মেই। শুধু তখন প্রোগ্রাম ফাইলটার নাম হবে 'gadha' নয়, 'gadha.exe' — উইনডোজ-এর ওটাই নিয়ম। '.exe' নামক ফাইল-টাইটেলটা বোঝাচ্ছে যে এটা একটা চালনীয় বা এক্সিকিউটেবল ফাইল, মানে প্রোগ্রাম। সরাসরি একে চালানো যাবে, কমান্ড প্রম্পট থেকে বা ইঁদুরের পেট টিপে আদেশ দিয়ে। এবার যদি এই প্রোগ্রাম ফাইল, সে গ্নু-লিনাক্সের 'gadha' হোক বা উইনডোজের 'gadha.exe', কোনো

সরল কমান্ড এডিটর দিয়ে খোলেন, তাতে আবার দেখবেন সেই অবোধগম্যতা, কিন্নর গন্ধর্বদের প্রাইমারির সিলেবাসে অবশ্যপাঠ্য সেই মেশিন ল্যাংগুয়েজ। আমরা বুঝব না, কিন্তু মেশিন বুঝবে। এবার আপনি ওই প্রোগ্রামটাকে সরাসরি চালান, লিনাক্স হলে ./gadha, বা উইনডোজ হলে gadha.exe — আদেশের চেহারাটা এরকম আলাদা হল কেন আমরা পরে বুঝব — সরাসরি ঠিক তাই করবে যা আমরা তাকে করতে বলেছিলাম, মানে, সেই সমূহ সম্মাননায় সমৃদ্ধ লাইনটিকে ফুটিয়ে তুলবে স্ক্রিনে, 'কিরে গাধা!!!'। মানে, আমাদের কম্পাইলার মূল কোড ফাইলটাকে কম্পাইল করে কম্পিউটারবোধ্য করে তুলল। যাতে এবার নির্দেশমালাটাকে কম্পিউটার বুঝতে এবং পালন করতে পারল।

৭।। অ্যাপ্লিকেশনস

হার্ডওয়ার থেকে সফটওয়ার অব্দি কম্পিউটারের গোটা কাজের এলাকাটার স্তরবিন্যাস দেখিয়ে আমাদের দেওয়া সারণীর ৪ নম্বর স্তরে অপারেটিং সিস্টেম একা, আর ৫ নম্বর স্তরে তিনটে উপাদান কমান্ড ইন্টারপ্রিটার, এডিটর এবং কম্পাইলার। এখানেই শেষ মূল প্রোগ্রাম কাঠামো বা সিস্টেম প্রোগ্রামের এলাকা। এর পর শুরু প্রয়োগ বা অ্যাপ্লিকেশনের এলাকা। যে এলাকায় মেশিন ঘিরে বসে ব্যবহারকারীরা তাদের নানা প্রয়োজনের নানা প্রয়োগমূলক প্রোগ্রাম বা অ্যাপ্লিকেশন সফটওয়ার চালায়। এখানে আমরা তিনটে উদাহরণ দিয়েছি, অ্যাকাউন্টিং সফটওয়ার, রেলওয়ের টিকিট বন্টন ব্যবস্থা আর ইন্টারনেটে ওয়েব ব্রাউজিং। এরকম আরো বহু সফটওয়ার আছে। এটাই হল সেই স্তর যেখানে সচরাচর বেশিরভাগ ব্যবহারকারী বা ইউজার কাজ করে।

এই জায়গায় একটা কথা বলার আছে। উইনডোজ জগতে এই ব্যবহারকারী বা ইউজার এবং পরিচালক বা সুপারভাইজর — এই দুটো আলাদা জায়গা খুব একটা স্পষ্ট নয়। কিছুটা আছে উইনডোজ এক্সপিতে। আর গ্নু-লিনাক্স তথা যে কোনো ইউনিক্স সিস্টেমেই এই ব্যাপারটা সর্বব্যাপী। এটাই এই মেশিনগুলোর নিরাপত্তার উৎস। কেন কোনো গ্নু-লিনাক্স সিস্টেমে কোনো ভাইরাস ইনফেকশন সম্ভব না, একদম কাঠামোগত ভাবেই, সেটা বুঝতে গেলে আমাদের এটা বুঝতে হবে। এটা নিয়ে বিশদ আলোচনা আসতে বহু দেরি আছে। সাত বা আট নম্বর দিনের আগে নয়। কারণ, তার আগে, ফাইল তথা ফাইলসিস্টেম তথা তার অধিকার এবং অনুমতির কাঠামো নিয়ে বহু কথা বলার আছে, যার উপর দাঁড়িয়ে আছে ইউনিক্স তথা যে কোনো গ্নু-লিনাক্স ব্যবস্থার গোটা কার্যধারাটা।

ইউনিক্স জগতে, সেটা ব্রান্ডনেম ব্যবসায়িক ইউনিক্স হোক, আর ওপন-সোর্স ইউনিক্স মানে গ্নু-লিনাক্স হোক, এই পরিচালক হল সর্বব্যাপী একচ্ছত্র ক্ষমতার অধিকারী, সিস্টেমের আভ্যন্তরীন প্রতিটি প্রত্যেকটি কিছুই তার ইচ্ছাধীন। এর নাম রুট। এবং ব্যবহারকারীরা ঠিক ততটুকু অধিকারই উপভোগ করে যতটুকু রুট তাকে অনুমতি দেয়। আমরা পরে ধীরে ধীরে এর সমস্ত জটিলতাগুলো বুঝতে পারব। এখন শুধু এটুকুই মাথায় থাক যে সাধারণ ব্যবহারকারী বা ইউজারের এলাকার সঙ্গে রুট বা সুপারভাইজরের অধিকারের একটা নাটকীয় তফাত আছে।

অপারেটিং সিস্টেমের স্তরের অন্য নাম কারনেল স্তর। এই কারনেল স্তর গোটাটাই সুপারভাইজরের এলাকা। সাধারণ ব্যবহারকারীর এখানে কোনোরকম কোনো প্রবেশাধিকারই নেই। তার আওতায় পড়ে এই অ্যাপ্লিকেশন সফটওয়ারগুলো ব্যবহার করা, তাও তার খুঁটিনাটি নিয়ন্ত্রকগুলো লেখা থাকে যে কনফিগারেশন ফাইলে, যে ফাইল অনুযায়ী অপারেটিং সিস্টেম এদের কাজ করার ব্যবস্থা করে দেয়, বিভিন্ন হার্ডওয়ার উপাদানগুলো এদের ব্যবহার করার সুযোগ করে দেয়, সেই কনফিগারেশন ফাইলগুলোও বদলানোর কোনো অধিকার থাকে না একজন সাধারণ ব্যবহারকারী বা ইউজারের। সাধারণ ব্যবহারকারী কমান্ড এডিটর দিয়ে প্রোগ্রাম লিখতে পারে, কম্পাইলার দিয়ে সেটা কম্পাইলও করতে পারে, কিন্তু কোনো সিস্টেম প্রোগ্রামে হাত দিতে পারে না। মানে সিস্টেম কাজ করার জন্যে যে প্রোগ্রামগুলো প্রয়োজন পড়ে। আমাদের টেবিল মোতাবেক বলতে গেলে, ৫ নম্বর স্তরে এসে কম্পাইলার এডিটর কমান্ড ইন্টারপ্রিটারদের নিয়ে সুপারভাইজর বা রুট-এর একচ্ছত্র অধিকারের এলাকা শেষ। এক থেকে পাঁচ অব্দি কারনেল মোড। কোনো ব্যবহারকারীর কোনো রকম স্বাধীনতা এই অব্দি বরাদ্দ নয়। কোনো সাধারণ ব্যবহারকারীর তৈরি করা কোনো প্রোগ্রাম সিস্টেম কোনো প্রোগ্রামে তথা কনফিগারেশনে তথা ফাইলে হাত দিতে পারেনা। কেন পারেনা এটা বোঝার জন্যে আমাদের ছয় নম্বর দিন অব্দি অপেক্ষা করতে হবে, যখন আমরা একটা

ফাইলের উপর মালিকানা তথা পড়ার/ লেখার/ চালানোর এই তিন রকম অধিকারের কাঠামোটা জানতে পারব। এটাই একটা গ্নু-লিনাক্স ব্যবস্থার সমস্ত নিরাপত্তার উৎস।

এর পর শুরু ইউজার বা ব্যবহারকারীদের এলাকা। বা শেল মোড। যারা বিভিন্ন অ্যাপ্লিকেশনকে প্রয়োগ করে বিভিন্ন কাজে। বা অকাজেও, যা তারা করতে চায়। লেখা থেকে মিউজিক বা ভিডিও এনকোডিং, গেমস থেকে থ্রি-ডি রেন্ডারিং, স্যাটেলাইট সুরভেইলান্স থেকে বিদ্যাচর্চার সর্বব্যাপী উন্মুক্ততা। নতুন প্রোগ্রাম লেখা থেকে শুরু করে পুরোনো প্রোগ্রামের চুল চিরে দেখা — যার অন্য নাম ডিবাগিং। এই প্রয়োগমূলক প্রোগ্রামের একটা বড় এলাকাকে আমরা আমাদের এই পাঠমালাতে আনবই না, যারা গুই বা গ্রাফিকাল-ইউজার-ইন্টারফেস বা ছবি-মাউসের জগতের অ্যাপ্লিকেশন। এই পাঠমালায় আমরা একটা গ্নু-লিনাক্স ব্যবস্থাকে শুধু তার কমান্ড মোডেই ধরার চেষ্টা করব।

আমরা আমাদের টেবিলে আলাদা আলাদা স্তরগুলোকে খুব স্পষ্ট ভাবে ভাগ করেছি। সবসময় অবশ্য সবগুলো স্তরকে এমন জল-অচল কামরায় আলাদা করা যায়না। যেমন রিয়াল-টাইম এমবেডেড সিস্টেমগুলো, ধরুন ক্রেডিট কার্ড বা সেলফোন ইত্যাদি। সেখান অপারেটিং সিস্টেম এবং অ্যাপ্লিকেশন এরকম স্পষ্ট ভাবে বিভক্তই নয়। বা এমনকি কোথায় ইউজার তথা অ্যাপ্লিকেশনের এলাকা শুরু এবং কোথায় সিস্টেম প্রোগ্রামের এলাকা শেষ তা নিয়েও চুলচেরা বিচারে অনেক তর্ক উঠতে পারে। কিন্তু সেগুলো আমাদের এই পাঠমালার আওতায় কোনোভাবেই পড়বে না।

অপারেটিং সিস্টেমের ভিত, এবং তার উপর দাঁড়িয়ে কম্পিউটারের গোটা কাজের কাঠামোটার সম্পর্কে একটা প্রাথমিক ধারণা হল আমাদের। এক নম্বর দিন শেষ হল। এরপর, দুই নম্বর দিন, আমরা সিপিইউ এবং তার চারপাশে ঘিরে থাকা নানা ভৌত উপাদানগুলোর সম্পর্ক নিয়ে প্রাথমিক একটা ধারণা তৈরি করব। তিন আর চার নম্বর দিনে আসবে হার্ডওয়ার এবং সফটওয়ারের দুটো নিকট দুনিয়াকে মিলিয়ে প্রাগইতিহাস থেকে ইতিহাসে কম্পিউটারের বিবর্তন। পাঁচ নম্বর দিন থেকে আমরা পুরোদমে একটা জ্যান্ত গ্নু-লিনাক্স সিস্টেমে ঢুকব।

glt-mad@ilug-cal.org

MADHYAMGRAM

গ্নু-লিনাক্স ইশকুল

জিএলটি মধ্যমগ্রাম
glt-mad@ilug-cal.org

শূন্য নম্বর দিনে আমরা হাতের সামনে একটা আস্ত পিসি পেয়ে নাড়াচাড়া শুরু করেছিলাম, ভৌত উপাদান নিয়ে একদম প্রাথমিক দু-চারটে কথা, তারপর একটা কম্পিউটারের তথ্য নাড়াঘাঁটা ও জমানোর তরকিব নিয়ে কিছু কথা বলে শেষ করেছিলাম। এক নম্বর দিনে আমরা শুরু করেছিলাম একটা কূট এবং আপাতঅগ্রাহ্য সেকশন দিয়ে, যাতে সিপিইউ-র রেজিস্টার কাঠামো এবং মেমরি পড়ার সিপিইউ-প্রথা নিয়ে কথা বলেছিলাম, তারপর গেছিলাম একটা কম্পিউটারের সফটওয়্যার অংশটার একটা কাঠামো খাড়া করার কাজে। প্রোগ্রাম, সফটওয়্যার, অ্যাপ্লিকেশন — এবং এদের সামগ্রিক আধার — মানে অপারেটিং সিস্টেম। অপারেটিং সিস্টেমের ভিতর সফটওয়্যারের বিভিন্ন ধারাগুলোর, কমান্ড ইন্টারপ্রিটার, এডিটর, কম্পাইলার, এবং অ্যাপ্লিকেশন — এদের ঠিক ভূমিকাটা কী সেই বিষয়ে একটা আন্দাজ দিয়েছিলাম। আজ, মানে জিএলটি-ম্যাড পাঠমালার দুই নম্বর দিনে অপারেটিং সিস্টেমের ভূমিকা নিয়ে একটু কথা বলে চলে যাব, সিপিইউর পরই কম্পিউটারের সবচেয়ে গুরুত্বপূর্ণ দিক — আইও মানে ইনপুট/আউটপুট নিয়ে প্রাথমিক কিছু কথায়। এই পরের অংশের অনেকটাই ঠিক এক নম্বর দিনের এক নম্বর সেকশনের মত কূট এবং আপাতঅগ্রাহ্য। পরে যখন শেল কারনেল নিয়ে দ-য়ে পড়বেন তখন বুঝবেন কত কাজের ওগুলো, ফিরে আসতে হবে। এই পাঠমালার মূল পরিকল্পনাটা প্রতি মুহূর্তে আমূল বদলে যাচ্ছে, খসড়াগুলো রয়েছে নিরন্তর প্রব্রজ্যায়, এই বদল ঘটানোর চক্করে ভাঙা ঠ্যাং-এর প্লাস্টারের নিচে হিলের উপর টাল খেতে খেতে আমায় বারকয়েক করে ঠাকুরঘরে যেতে হচ্ছে, এক্সট্রা, ঠাকুরঘরে ছাড়া আমি আবার খুব মন দিয়ে কিছু ভাবতে পারিনা। ঠাকুরঘর শব্দটা, প্লিজ, এখানে একটু ভিন্ন কনটেক্সটে পড়বেন — যে কলোনিয়াল সভ্যতার দায় নিয়ে রামরাম বসু মশাই তখন সেবারত, আমাদের আদি পরিত্রাতাতাদের প্রমুখ উইলিয়ম কেরিকে, তখন শীতের ভোরের কুয়াসা ফেটে আলো ক্রমে আসিতেছে, রক্তিমতা প্রভাব বিস্তার করিবে, আমরা কলোনির প্রাকৃতজনেরা পশ্চিমের সভ্যতার পুঁজির মাইক্রোসফটের দিকে ক্রমে অগ্রসরমান পুষ্পের উষ্ণতায় চিহ্নিত হইব, নদীবুকে ভাসমান বজরায়, নদীতটরেখা জুড়ে সারিসারি উপাসীন মানবমিছিল দেখে কেরি প্রশ্নশীল, প্রতিটি উপাসীন মানুষের সামনে উপাসনার উপচার, উজ্জ্বল ধাতুপাত্রে পূতবারি, রামো রামো, রামরাম বললেন, দে আর প্রেয়িং টু দেয়ার গড স্যার, আমিও তো আজো বহন করে চলেছি সেই কলোনিয়ালতার সভ্যতার আলোকপ্রাপ্তির উত্তরাধিকার।

।। দিন দুই ।।

১।। দুটো কাজ : করছে এক জন

আগের দিন অপারেটিং সিস্টেম নিয়ে বেশ কিছু কথা বলার পরও, আজ যদি প্রশ্ন করা যায়, অপারেটিং সিস্টেম বস্তুটা ঠিক কী — খায় না মাথায় দেয় — আমাদের খুব স্পষ্ট একমুখী একটা দ্বর্থহীন উত্তর দিতে বেশ ব্যথা হবে। শুধু তাই নয়, অনেকদিন ধরে কম্পুটার ব্যবহার করার পর নিজের থেকেই বেশ অনুভব করা যায় কাজের ক্ষেত্রের গোটাটা জুড়ে অপারেটিং সিস্টেমের সর্বব্যাপী সর্বত্রগামী সর্বশক্তিমান উপস্থিতি, কেউ একটা অন্তরালে বসে সব কিছু গুছিয়ে যাচ্ছে। সুগৃহিণীপনার হদ্দমুদ্দ একেবারে। কে কী করতে পারে, করতে চাইতে পারে, সচরাচর করে থাকে, কে কে কখন কখন কোথায় কোথায় কী করছে, এই করার সময় তাদের কী কী দরকার পড়ে, সেই দরকারগুলোকে আগে থেকে ভাগবাঁটোয়ারা করে রাখা, টেনির হাত থেকে ছিনিয়ে খেতে পারেনা পিলেবান প্যালা, তার জন্যে একটু কিছু তুলে রাখা। ইত্যাদি। তাকে বোঝা যায় সদাসর্বদা, কিন্তু চিহ্নিত করার কাজটা খুব সমস্যার হয়।

এই উপস্থিতিটা আরো প্রকট হয় গ্নু-লিনাক্স সিস্টেমে। প্রতি পদে পদে। এমনকি বিরক্ত হয়ে সিস্টেম বন্ধ করে দিতে চাইলেন, টাইপ করে কমান্ড দিলেন, পাওয়ারঅফ (poweroff), তাতেও ও সঙ্গে সঙ্গে জানাবে, তোমাকে খোকা এই কাজটা করতে গেলে রুট হতে হবে। হার্ডডিস্কের কোনো একটা পার্টিশনে কোনো একটা ফাইল লিখতে বা পড়তে চাইলেন, আপনাকে জানাল, না খোকা, ওটা এখন মাউন্ট করা নেই, মাউন্ট করতে চাইলেন, আপনাকে জানাল, তোমাকে রুট হতে হবে। পার্টিশন কী তা আমরা পরে জানব, পার্টে পার্টে এগোতে হবে না? মাউন্ট ব্যাপারটাও তাই, ইস্টবেঙ্গল মাঠের র‍্যামপার্টের নালার পাশে তাড়া করে আসা দামী চকচকে ঘোড়ার পিঠে শুঁটকো কনিষ্ঠবলের মানে মাউন্টেড পুলিশের হাতের ব্যাটনের চেয়ে অনেক কম জটিল একটা বস্তু। বেশ শিখে যাব আমরা, মানে শিখতে

আমাদের হবেই। রুটের এই যত্রতত্র রুট বন্ধ করে দিয়ে ট্রাফিক থামিয়ে মন্ত্রীর মহামন্ত্রীর নাম নেওয়ার অত্যাচারটা মাঝে মাঝে রীতিমত বেদনার হয়ে ওঠে, গোটা ব্যাপারটা ঠিক করে না-বুঝলে। অজস্র কমান্ড আছে যেগুলো শুধু রুটই চালাতে পারে, আপনি কমান্ডটা টাইপ করে এন্টার মারুন, আপনাকে শুনিয়ে দেবে, কমান্ড খুঁজে পাইনি, নট ফাউন্ড। অথচ সু হয়ে নিন, কমান্ড প্রম্পটে সু (`su` — **S**uper-**U**ser/**S**ubstitue-**U**ser) কমান্ড দিয়ে, দেখবেন সেই একই কমান্ড চমৎকার চলছে। সু হতে গেলে আপনার অবশ্য রুট পাসওয়ার্ড জানতে হবে, কারন সিস্টেম সেটা চাইবে।

এগুলো আসলে গত দিনের একদম শেষে আমাদের বুড়ি ছুঁয়ে চলে আসা নিরাপত্তার গল্পেরই পার্ট, গ্নু-লিনাক্স তথা যে কোনো ইউনিক্স সবসময়েই, আপনা থেকেই, যার টেকনিকাল নাম ডিফল্ট, সেই ডিফল্টেই ধরে নেয় যে, সিস্টেমটা একটা নেটওয়ার্কে আছে। মাল্টি-ইউজার তো বটেই, মাল্টি-মেশিনও। সেখানে একজন সুপারভাইজর বা সুপারইউজার বা রুটের একচ্ছত্র স্বৈরতান্ত্রিক উপস্থিতি অত্যাবশ্যক। কিন্তু আমরা যারা স্ট্যান্ড অ্যালোন মানে একা একলা একটা পিসিতে কাজ করি, তাদের প্রায়ই খাজনার চেয়ে বাজনা বেশির একটা অনুভূতি তৈরি হয় এতে। কিন্তু পরে দেখব আমরা, একটা কদাচিৎ কখনো ডায়াল আপ মোডেম দিয়ে নেটনিষিক্ত, মোটের উপর একা একলা একটা টেবিলে বিরাজ করা একটা পিসিতেও কতটা জরুরি হয়ে ওঠে এই নিরাপত্তার ব্যাপারটা, মূলত ভাইরাস সংক্রমণের হাত থেকে বাঁচতে। আমি নিজে গ্নু-লিনাক্সে এসেছিলাম এই ভাইরাস থেকে বাঁচতে। আমার বাড়িতে একটা স্টাডি সার্কলে আসত তথাগত, পরপর ভাইরাসে আমাকে জেরবার দেখে বলেছিল, তুমি লিনাক্সে চলে এসো দীপঙ্করদা, তখন দুচারটে করে ভাইরাস ফোল্ডারে ফোল্ডারে পুষতেও পারবে, যদি চাও — আমি নিজে তো একটা ভাইরাস সংগ্রহ তৈরি করব ভাবছি — লিনাক্সে ওরা সম্পূর্ণ অকেজো, সেই ভাবেই বানানো অপারেটিং সিস্টেমটা। পরে আমরা দেখব ভাইরাসে এই অসংক্রমণীয়তাটা গোটাটাই এই নিরাপত্তা ব্যবস্থার অংশ।

কিন্তু মুহূর্মুহূ যখন সিস্টেম খুঁতখুঁত করতে থাকে, এটা কোরোনা, ওটা ছুঁয়োনা, ওদিকে তাকিয়ো না, তখন সত্যিই বেশ বিরক্ত লাগে। এর থেকে বাঁচার একটা উপায় হল, সিস্টেমের `su`-জন হওয়ার হদিশ, বাঁশরী-সঙ্কেত, মানে রুট পাসওয়ার্ডটা মনে রেখে দেওয়া, শুধু কুজন মানে ইউজার থেকে প্রায়ই চলেনা, কূজনটুকুও চলেনা প্রায়ই। এর থেকে বাঁচার কিছু উপায় কোনো কোনো গ্নু-লিনাক্স ডিস্ট্রিবিউশন ইনস্টলেশনের সময় দিয়ে থাকে — কী চাও বাছা, এমন ব্যবস্থা করে দেব যে কোনো ব্যবহারকারীকেই সিস্টেম রুট বলে ভাববে? কিন্তু এর মানে, গ্নু-লিনাক্স সিস্টেমের সবচেয়ে জোরালো একটা জায়গাকে আপনি প্রথমেই বিসর্জন দিয়ে এলেন। শুধু আপনি কেন, আক্রমণকারী ভাইরাস যখন চলতে চাইবে, ভাইরাস-রা হল একরকমের প্রোগ্রাম, যারা সঙ্গত ব্যবহারকারীর ইচ্ছে অনিচ্ছের তোয়াক্কা না-রেখেই চলে, এবং সিস্টেম জুড়ে অবাঞ্ছিত খুনখারাপি ঘটাতে থাকে, তখন সেই ভাইরাসকেও সিস্টেম রুট বলে ভাববে, এবং নিজের অন্দরতম অভ্যন্তরও ওয়াইড ওপেন করে দেবে তার সামনে, যেটা শুধু রুটের সঙ্গেই করার কথা তার।

এই পাঠমালার পরের একাধিক দিনে এই নিয়ে আলোচনা তুলব আমরা, গ্নু-লিনাক্স কাঠামোর এই নিরাপত্তা ব্যবস্থা নিয়ে। যার মূল উপাদান হল একটা ফাইল পড়ার বা দেখার বা চালানোর অধিকার, এবং একটা ডিরেক্টরিতে ঢোকার বা সেখানে কী কী ফাইল আছে সেটা দেখার অধিকার, এবং ফাইল বা ডিরেক্টরির উপর মালিকানা। বহু ডিরেক্টরিতে আপনি ঢুকতেই পারবেন না, কারণ সেই ডিরেক্টরিতে ঢোকার অনুমতি নেই আপনার। বহু ফাইল চালানোর, বদলানোর অনুমতি তো ছেড়েই দিন, এমনকি পড়ার অনুমতিই নেই আপনার। সুজন-সঙ্কেত না দিলে, সেই ফাইলে কী আছে ফাইলে সেটাই দেখতে পাবেন না আপনি। নিরাপত্তার খ্যাঁচটা এখানেই — আপনার যেমন নেই, ভাইরাসেরও নেই। অনেকটা নিজের মাত্রা টেনে ভাইরাসের যাত্রাভঙ্গ। কোনো সিস্টেম ফাইলই বদলানো তো দূরের কথা পড়ার অনুমতিই যদি না থাকে তার, তাহলে সে বেচারা আর কী করবে, বসে বসে ভাইরাসান্ডা ভাজা ছাড়া? শুধু তথাগতর না, এখন আমার সিস্টেমেও শত ভাইরাস বিকশিত হোক, শত না হোক নিদেন দু-চারটে ভাইরাস। আপনারা কেউ চাইলেই আনন্দের সঙ্গে দিতে পারি, আমার মন খুবই নরম।

ভাইরাস একটা গ্নু-লিনাক্স সিস্টেমে একমাত্র তবেই সংক্রমণ করতে পারে যদি আপনি রুট হয়েই ভাইরাসটাকে সক্রিয় করেন। তাতেও অবশ্য কিছু জাটিল্য আছে। এই পাঠমালার একদম শেষ দিকে মানে আট নয় আর দশ নম্বর দিনে, যখন আমরা গ্নু-লিনাক্স ফাইলব্যবস্থা আর কনফিগারেশন নিয়ে আলোচনা করব, তখন বুঝবেন। যাইহোক,

ওরকম আত্মপিসিঘাতী হওয়ার বাসনা থাকলে তার সহজতর অনেক পন্থা আছে। আর এরকম ঘাতক হবেন কেন, হিংসা পরম অধর্ম, পরম ঘৃণা এবং তাচ্ছিল্যের সঙ্গে আপনার পিসিটাকে জিএলটিতে দিয়ে দিন, একটা বাড়তি মেশিন রাখতে পারলে বেশ সুবিধে হত। অনেকেরই বেশ সুবিধে হত।

যাকগে, যেখান থেকে শুরু করেছিলাম, অপারেটিং সিস্টেমের মার খুব খাওয়া যায়, কিন্তু ঠিক কোথা থেকে মারছে সেটা খুব স্পষ্ট করে চিহ্নিত করা যায়না। অনেকটা আমাদের এই স্বাধীন তৃতীয় বিশ্বের মত। আগে ব্যাপারটা ছিল স্পষ্ট চিহ্নিত, তারাপদ রায়ের কুকুরের গলার বকলসে যেমন লেখা থাকত, ‘আমি পণ্ডিতিয়ার তারাবাবুর কুকুর, আপনি কোন বাবুর কুকুর?’, বাবুত্বটা সেখানে নিশ্চিত এবং নির্দিষ্ট, অঙ্কের ভাষায় বললে, ফাইনিট এবং ডেফিনিট। কোনো দেশের ছিল ইংরেজ বাবু, কারোর ফরাসি বাবু, এরকম, এখনও, প্রতিমুহূর্তেই, আঙিনা জুড়ে বাবুর পদচিহ্ন পাই, বাবু তো আছেই, কিন্তু ঠিক কোন বাঁধাবাবু আমায় রেখেছে সেটা বলা শক্ত, রোজই এত লোককে ঘরে বসাতে হচ্ছে, নানা এমএনসির নানা ব্রান্ডের নানা লোক। ঔপনিবেশিক বা উত্তরঔপনিবেশিক, বাবু আমাদের আছেই, শুধু তাকে দেখানো যাচ্ছেনা আলাদা করে। ঠিক সেরকম সব কিছুই তার হাতে, সব হেঁসেলের চাবি, সব শিকের হদিশ, কিন্তু তাকে আলাদা করে টেনে এনে দেখানো খুব শক্ত। এই হেঁয়ালির একটা বড় কারণ এই যে, একদম শুরু থেকেই অপারেটিং সিস্টেম এমন দুটো কাজ একই সঙ্গে করে চলেছে যাদের মধ্যে সরাসরি কোনো সম্পর্ক নেই, মানে একটা কাজ অন্য কাজটার উপর সরাসরি নির্ভরশীল নয়। আরো দুটো কাজই এমন শক্তিশালী এবং প্রবল রকমে তার উপস্থিতি চাগিয়ে রাখে সিস্টেমে, যে একটা কাজের আইডেন্টিটির মধ্যে অন্যটা ঢুকে আসে, মোটের উপর গোটাটাই ঘেঁটে দেয়।

> কাজ নম্বর এক। মেশিনটাকে মেশিনের বাইরে বাড়িয়ে তোলা, পরিবর্ধিত মেশিন করে তোলা। ভৌত মেশিনের সাথে একই সঙ্গে তাকে সাঙ্কেতিক মেশিন ভৌতিক মেশিন করে দেওয়া। দেরিদাদা যেমন বলেছিলেন ইউরোপের বাইরে ইউরোপের বেড়ে চলার কথা। ঠিক ভৌত উপাদানগুলোকে মিলিয়ে মেশিনের যে ভৌত উপস্থিতি, তার বাইরে মেশিনকে বাড়িয়ে তোলা — এক্সটেন্ডিং দি মেশিন।
>
> কাজ নম্বর দুই। মেশিনের এই তুমুল বাড়াবাড়ির আমলে, ভৌত মেশিনকে সাঙ্কেতিক তথা ভৌতিক তথা ভারচুয়াল করে দেওয়ার সঙ্গে সঙ্গে মেশিনের ভৌত উপাদানগুলোর যে সম্মিলিত রসদ বা রিসোর্স — তার প্রয়োজনমাফিক সঠিক ব্যবহারের ব্যবস্থা করা। রসদগুলো যাতে বাজে খরচ না-হয় তার পাহারা রাখা। সিস্টেমের আদত রসদগুলো এলোমেলে হয়ে যেতে না-দেওয়া, ঘেঁটে যেতে না-দেওয়া। রিসোর্স ম্যানেজমেন্ট।

আর ঘটনাটা এই যে, অপারেটিং সিস্টেম নিয়ে কে বলছে এবং কেন বলছে তার উপর দাঁড়িয়ে, কোন অবস্থায় কোন প্রয়োজনে বিচার করছি, তার উপর নির্ভর করে, কখনো এই কাজটা বেশি গুরুত্ব পায়, কখনো ওইটা, কিন্তু দুটো কাজ একসঙ্গে কিছুতেই পিকচারে আসে না। কাজ দুটোকে একটু গুছিয়ে আলোচনা করা যাক।

২।। মেশিন মেশিনতর মেশিনোত্তর — অপারেটিং সিস্টেম মানে পরিবর্ধিত মেশিন

যেমন আগেই বলেছি আমরা, মেশিনের যে ভৌত গঠন বা আর্কিটেকচারটা খুব আদিম জংলি টাইপের, একদম রক্তমাংসের লেভেলে খুব টুকিটাকি কিছু ছাড়া তাকে শেখানো খুব শক্ত। আর্কিটেকচার বলতে এখানে আমরা ঠিক কী কী বুঝি তার একটা তালিকা তৈরি করে ফেলা যাক। এদের উপরেই দাঁড়িয়ে থাকে একটা কার্যরত চালু মেশিনের অপারেটিং সিস্টেম থেকে অ্যাপ্লিকেশন অব্দি হার্ডওয়ার আর সফটওয়ারের গোটা ইমারতটা।

(১) প্রাথমিক আদেশমালা বা ইন্সট্রাকশন সেট।

> মেশিনে বিদ্যুৎ অন করার পরই একদম প্রাথমিক চালু হওয়ার বা বুটের সময়, মানে মেশিন যখন ক্রমে ক্রমে কাজ করার উপযুক্ত অবস্থায় আসছে, তখন কী করতে হবে সেটা কম্পিউটারকে বলে দেয় এই ইন্সট্রাকশন সেট। এর উদাহরণ বেসিক ইনপুট আউটপুট সিস্টেম বা বায়োস (BIOS — **B**asic-**I**nput-**O**utput-**S**ystem)। কালো স্ক্রিনে যখন লেখা ফুটে উঠছে লাইন লাইন করে সাদা অক্ষরে, দেখবেন একদম নিচের দিকে রয়েছে, সেট-আপে যদি ঢুকতে চাও ডেল মারো — ‘Hit DEL to enter Setup’। এই ‘সেট-আপ’ বায়োস সেট-আপ, বেসিক ইনপুট আউপুট বা তথ্য ঢোকানো বার-করার প্রাথমিকতম কায়দাকানুনগুলো কম্পিউটারকে

বলে দেয় এই বায়োস। বায়োসের বিভিন্ন ব্যবস্থা বা সেটিং-কে বদলানো যায় বায়োস সেট-আপ-এ ঢুকে। সেটা করতে গেলে আপনাকে ওই 'DEL' চাবিটা টিপতে হবে, এবং টিপতে হবে তখুনি, সেই মুহূর্তেই। দেরি করলেই কম্পিউটার সম্পূর্ণ বুট করে যাবে।

এবার ভাবুন, এই যে 'ডিলিট' বা 'ডেল' বা 'DEL' সুইচটা টিকলে এটা করতে হবে, এই কথাটা কম্পিউটার জানছে কী করে? শূন্য নম্বর দিনে করা আলোচনাটা মনে করুন, নিয়ন্ত্রণ তথা সিদ্ধান্ত নেওয়ার অধিকার যার, সেই সিপিইউ জানছে কী করে, যে ওই সুইচটা টিপলে এটা করতে হয়? তাকে কে শেখাল? কী ভাবে শেখাল? অন করলাম আমরা, তার আগে মেশিনটা অফ ছিল, কোনো কাজ করছিল না, সিপিইউতে কিছু নেই, র‍্যামে কিছু নেই, আর হার্ডডিস্ক থেকে পড়তেও শুরু করেনি তখনো, পরে দেখবেন, বুটপ্রক্রিয়াটা আরো কিছুদূর চলার পরে কম্পিউটার হার্ডডিস্ক থেকে পড়তে শুরু করে, তাহলে এটা করতে কম্পিউটার শিখল কী করে? এই শেখানোটা তৈরি করে দিচ্ছে প্রাথমিক আদেশমালা বা ইন্সট্রাকশন সেট। এটা লেখা থাকে রম চিপে, যার কথা আমরা আগেই বলেছি। বলা থাকে, মেশিন অন হল, তোমার মধ্যে বিদ্যুৎ বা প্রাণ এল, তো ইহা ইহা করিতে আরম্ভ করো, শিশুগণ দাও মন নিজ নিজ পাঠে। মেশিনও বুঝে গেল, এখন এই এই লেখা ফুটিয়ে তোলো স্ক্রিনে, যার মধ্যেই পড়ে ওই ডেল টিপলে বায়োস সেট-আপে ঢোকার নেমন্তন্ন। এই প্রাথমিক ইন্সট্রাকশন সেটে যদি থাকত, মেশিন বুট হওয়ার পরেই ঘ্যাঁক করে একটা আওয়াজ করো, তাহলে সে তা-ই করত, তবে সেটা খুবই শক্ত হত, কারণ, ঘ্যাঁক আওয়াজটা একটা জটিল মিডিয়া ফাইল হত, সেই ফাইলকে চালানোর মত প্রোগ্রাম তার আগেই চালু করে দেওয়ার ব্যবস্থা করতে হত।

(২) সিস্টেমের নিজস্ব স্মৃতি-সংগঠন বা মেমরি অর্গানাইজেশন।

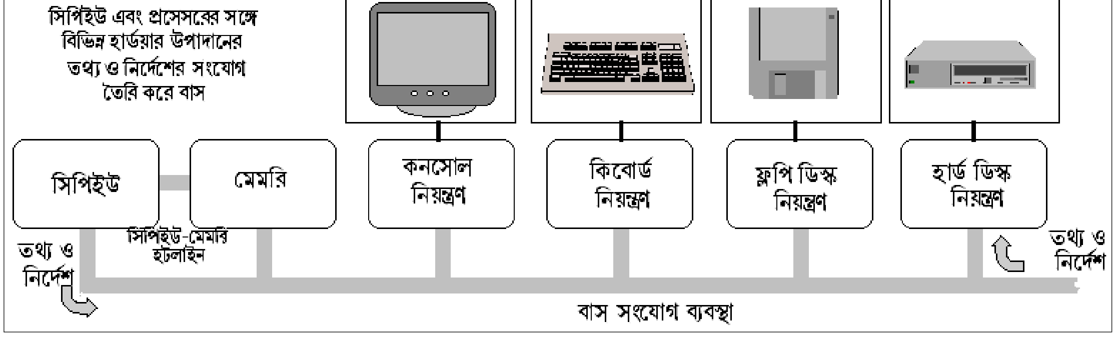
সিস্টেমের যে স্মৃতি-সংগঠন বা মেমরি অর্গানাইজেশন — সেটাও পড়ে এই আর্কিটেকচার বা ভৌত গঠনের মধ্যে। স্মৃতির কী কী ধরনের ভৌত রসদ ভরা আছে মেশিনে, কী কী পরিমানে, চলমান নানা প্রোগ্রামের মধ্যে তাদের কী ভাবে বণ্টন করবে অপারেটিং সিস্টেম — এই গোটাটা পড়ে এই স্মৃতি-সংগঠন বা মেমরি-অর্গানাইজেশনের মধ্যে। একদম প্রাথমিক আলোচনাটুকু আমরা আগেই করেছি। পরে আরো বহুবার আসবে আলোচনাটা, যখন যখন সিস্টেম বুঝতে গিয়ে দরকার পড়বে।

(৩) কম্পিউটারে তথ্য ঢোকানো এবং বার-করা মানে ইনপুট-আউটপুট বা আইও (I/O)।

আগেই বলেছি, কম্পিউটারের কাজ তথ্য চটকানো। সেই তথ্যকে চটকানোর আগে কম্পিউটারে ঢোকাতে হবে, এবং চটকানো শেষ হওয়ার পরে কম্পিউটার থেকে বার করতে হবে, একেই এক কথায় বলা হয় আইও। শূন্য নম্বর দিনে এই নিয়ে কিছু আলোচনা আমরা করেছি, পরে আরো আসবে। আজকেই, একদম শেষে গিয়ে আরো একটু আসবে।

(৪) কম্পিউটারের বাস-গঠন।

এই বাস বা BUS বলতে বোঝায় একটা কম্পিউটারের বিভিন্ন কেজো ইউনিটগুলোর ভিতর পারস্পরিক যোগাযোগটা যে পথ বেয়ে ঘটে — একটা বিদ্যুৎখচিত পথরেখা — ঠিক আমাদের প্রতিদিনের যাতায়াতের বাসের মতই, তথ্যকে যে কোনো জায়গা থেকে যে কোনো জায়গায় নিয়ে যায়। এই বাসের মধ্যে আবার নানা প্রকারভেদ হয়। নাম থেকেই আন্দাজ করা যায়, লোকাল বাস গুলো সিপিইউ বা সেন্ট্রাল প্রসেসিং ইউনিটের বিভিন্ন অংশগুলোর ভিতর যোগাযোগ রাখে — লোকাল দূরত্বে মানে কাছেপিঠে। আর দূর পাল্লার এক্সটার্নাল



> বাসেরা যোগাযোগ রাখে মেমরি এবং আনুষঙ্গিক যন্ত্রপাতি বা পেরিফেরালের সঙ্গে। আবার ঠিকানা জেনে রাখার এবং যোগাযোগের কাজ করে অ্যাড্রেস বাস। অ্যাড্রেস বাসরা খতিয়ান রাখে মেমরির বিভিন্ন এলাকার, কোথাকার তথ্য কোথায় গড়ায়, কোথায় কে যেতে পারে, বা যায়, বা আগে গিয়েছিল — তাই এখন আছে — মাঝরাত্তিরে বিনা ওয়ারেন্টে তুলে আনা যাবে, আরো যদি কলেজ-ইউনিভার্সিটির আবোধা মাস্টার হয়। এই তিন রকম, লোকাল বাস, এক্সটার্নাল বাস আর অ্যাড্রেস বাসদের নিয়ে তৈরি বাস-ব্যবস্থা, এই নিয়ে আরো আলোচনা আছে আমাদের আজই।

উপরের এই চার ধরনের উপাদান, ইন্সট্রাকশন-সেট, মেমরি-সংগঠন, ইনপুট-আউটপুট, আর বাস-গঠন — এদের নিয়ে তৈরি মেশিনের আদিম জংলি আর্কিটেকচারের স্তর। এই স্তরে সে এতটাই জংলি যে এখানে কম্পুবাবুকে কিছু শেখানো কদাকার রকমের শক্ত, বিশেষ করে যদি ঢোকানো-বার-করা বা ইনপুট-আউটপুট হয়।

আগের দিন মানে এক নম্বর দিনে, আমাদের সি-প্রোগ্রাম লেখা এবং রান করানোর কথাটা মনে করুন। সিকিতম সি না জেনেই, শুধু এই কম্পিউটারকে গাধা বলে ডাকার বিমল বিলাসে একটা রীতিমত দীর্ঘ্য শেষহীন নিরবচ্ছিন্ন রকমের লম্বা চারলাইনের gadha.c কোডফাইল লিখে, সেটাকে কম্পাইল করে প্রোগ্রাম বানিয়ে, সেই প্রোগ্রাম চালিয়ে আমাদের C-তল মানে আমেরিকান রকমের 'কুল' C-শিক্ষার প্রমাণ দিয়েছিলাম।

এবার ধরে নিন, ওই কোডটা লিখে, কম্পাইল করে প্রোগ্রাম বানিয়ে, তাকে চালিয়ে যা যা করেছিলাম আমরা, মানে 'কিরে গাধা!!!' বলে ডাকা — সেই একই কাজ করতে হবে আমাদের, কোডে লেখা আদেশগুলোই পালন করতে হবে, শুধু এখন আর কোনো অপারেটিং সিস্টেম নেই, কম্পাইলার নেই, কিচ্ছু নেই। শুধু আদিম এই আর্কিটেকচার বুকে নিয়ে কম্পিউটারটা দাঁড়িয়ে আছে। এই অবস্থায় দাঁড়িয়ে, ঠিক 'gadha.c' কোডের থেকে তৈরি 'gadha' প্রোগ্রামের কাজটাই করতে হবে আমাদের। মানে, ওই উপরে নিচে এক লাইন ফাঁকা রেখে ওই সম্মান সম্বোধনটাকে ফুটিয়ে তুলতে হবে স্ক্রিনে।

এবার ভাবুন, কাজটা শুরু হবে কোত্থেকে? প্রথমে ওই কোডফাইলের আদেশগুলো তাকে পড়াতে হবে, তারপর সেই অনুযায়ী তাকে কাজ করতে বলতে হবে। তার মানে মোট যত জিনিষ আপনার কম্পিউটারকে দিয়ে করাতে হবে, তার মধ্যে কী কী পড়বে তার কিছুটা ভাবার চেষ্টা করা যাক। সিস্টেমকে আমাদের নিঁখুত ভাবে বলে দিতে হবে, যে ডিস্ক থেকে আদেশগুলো পড়বে, সেই ডিস্ক মানে হচ্ছে এই সার্কিটের এই অংশ থেকে এই কানেক্টরের এই সংযোগ। সেখানে যাও। যে কৌটোটা পাবে ভিতরে মোটরটাকে অন করো, মানে এত ভোল্টের এতটা বিদ্যুৎ পাঠাও, ঘোরাও। এবার ঘুরতে থাকা ডিস্কের উপর কাঁটাটা নড়াও। এত এত নম্বর সেক্টর পড়ো। এটা করতে গিয়ে কোন ডিস্কের কোন সেক্টরে কোন কাঁটা কত ডিগ্রি কোণে ঘোরাবে তাও বলে দিতে হবে। তারপর বলে দিতে হবে, ওই আদেশমালাকে মেমরির এত এত নম্বর সেক্টরে এইভাবে তোলো। এই ভাবে বলতেই থাকো বলতেই থাকো, এতো সবে কলির সন্ধে। পুরো প্রক্রিয়াটা শেষ হবে স্ক্রিনে, মানে সার্কিটের এই অংশের এই সংযোগে এই এই পিক্সেল বা আলোকবিন্দু উদ্দীপ্ত করবার জন্যে তুমি এই এই পরিমাণ বিদ্যুৎ পাঠাও। শিউরে উঠবেন না, প্রায় এই ভাবেই কাজ করতে হত কম্পিউটারে, মাত্র দশক চারেক আগেই।

এবার দ্বিতীয় স্টেপ ভাবুন। আর একটা সিচুয়েশন। এবার কাজটা আর অতটা বীভৎস নয়। ধরুন কম্পাইলারটা আছে আমাদের কাছে। মানে, আমাদের সি-তে লেখা কোড বা আদেশমালাকে সে অনুবাদ করে দেবে কম্পিউটারবোধ্য মেশিনভাষায়, আমাদের ওই আদিম আর্কিটেকচার যা বোঝে। শুধু ইনপুট-আউটপুটের ডে খুঁটিনাটিগুলো আপনার হয়ে অপারেটিং সিস্টেম নিজেই বুঝে নেয়, সেটা সে আর বুঝছে না। তাই ইনপুট-আউটপুট বা আই-ও সংক্রান্ত ডিটেইলস আমাদের নিজেদেরই জানিয়ে দিতে হবে। মানে, এবারেও কাজটা ভাবুন, অপারেটিং সিস্টেমকে আপনার নিজেরই জানিয়ে দিতে হবে, যে ডিস্কে লেখা আছে প্রোগ্রামটা সেই ডিস্ক মানে ঠিক কোন ভৌত উপাদানটা, এবং সেখান থেকে তথ্য পড়া মানে ঠিক কী করা বোঝায় সেটাও বলে দিতে হবে আপনাকে। বলে দিতে হবে, স্ক্রিনে লেখা মানে কী, এবং স্ক্রিনের ভৌত উপাদানের খুঁটিনাটিও বলে দিতে হবে একই সঙ্গে। কম্পিউটারকে গাধা বলে ডাকার বাসনাটা যতই গাঢ় হোক, আপনার আর ইচ্ছে হচ্ছে প্রোগ্রামটা চালানোর?

অপারেটিং সিস্টেমের কাজ ঠিক এটাই। আপনার হয়ে অনেকটা কাজ করে রেখে দেওয়া। বিভিন্ন ভৌত উপাদানকে, তাদের অংশগুলোকে চিনে রাখা, তাদের নাম দিয়ে রাখা। তাদের কাজ করার খুঁটিনাটি নিজের মধ্যে ভরে রাখা, যাতে আপনার কাজের প্রয়োজন হওয়া মাত্রই, আপনি ঠিক যে ভাবে চান সেভাবেই পেতে পারেন সিস্টেমটাকে।

প্রোগ্রামচিদের চাই একটা সরল সহজ সোজাসাপ্টা ভূমি, তার নাম যাই হোক, '/home/user' বা '/mnt/floppy', কিম্বা উইন্ডোজে 'C:\' বা 'A:\'। ওই নামগুলোর পিছনে ভৌত জায়গাগুলোকে আলাদা করে চিনিয়ে দিতে হবেনা। সেখানে 'খোলো' বা 'open' বললেই ফাইল খুলে যাবে, কিবোর্ডে টাইপ করলেই নতুন ঝকঝকে সাইবারপ্রোজ্বল সুখিনী বর্ণমালাকে যোগ বিয়োগ করা যাবে, 'তুলে রাখো' বা 'save' বললেই সঞ্চয় হয়ে যাবে ফাইলটা। খোলা বা তুলে-রাখা মানে কী সেটা সিস্টেম নিজে নিজেই বুঝে যাবে। এমনকি টাকা-পয়সা রাখার মত ব্যাংক যাওয়ার ঝঞ্ঝাটটুকু অব্দি করতে হবেনা। অর্থাৎ মেশিনের ওই আদিম জংলি আর্কিটেকচারের প্রাকৃত জগত থেকে অনেক উন্নত এলিট আলোকপ্রাপ্ত একটা সাংকেতিক ভূমি — '/dev/hda1' বা 'C:\' নামে যাকে ডাকা বোঝা এবং বদলানো যাবে। গ্নু-লিনাক্সে বা উইনডোজে আমরা যেভাবে সচরাচর আমাদের হার্ডডিস্ককে বুঝিয়ে থাকি। এই ডাকাটা এখন সাংকেতিক এবং বিমূর্ত, সিম্বলিক এবং অ্যাবস্ট্রাক্ট।

এই ডিস্ক মানে এবার কিছু নির্দিষ্ট ডিরেক্টরি, এবং তাদের পেটের মধ্যে আবার আরো ডিরেক্টরি বা সাব-ডিরেক্টরি। আর ডিরেক্টরিগুলোকে ফাঁকা রাখার জন্যে তো আর আপনি অত পয়সা খরচ করে মেশিন কেনেননি। ডিরেক্টরি থাকলেই তাদের ভরে দিতে হয় ফাইল দিয়ে। তাই ডিরেক্টরির পর ডিরেক্টরি ভর্তি গুচ্ছ গুচ্ছ ফাইল। যার প্রতিটি ফাইলই আপনি চাইলে খুলতে পারেন, বদলাতে পারেন, বন্ধ করতে পারেন, তাদের নিয়ে কাজ করতে পারেন, আপনার নিজের পয়সায় কেনা মেশিন বলে কথা। এবং এতো অনামিকাকে অনাম্নীর লেখা চিঠি নয় যে 'আনজান লিখ্‌খা থা উপর, নিচে নেম ছে', অর্থাৎ তাদের প্রত্যেকটি ফাইলেরই একটা নাম আছে, যা দেওয়া যায়, বদলানো যায়, অন্য নামে একই ফাইলের প্রতিরূপ বানানো যায় বা কপি করা যায়। এখন থেকে আপনি শুধু ফাইলের নাম, ডিরেক্টরির নাম, আর ড্রাইভের নাম জানলেই যে কোনো ফাইলকে নিয়ে যে কোনো কাজ করতে পারবেন।

এই সাংকেতিক নাম দিয়ে নির্দিষ্ট ভূমিটা কিন্তু আর ভৌত মেশিন নয়। মেশিন মানে ছিল কিছু বৈদ্যুতিক তার, কিছু সিলিকন চিপ, কিছু ধাতুর চাকতি, কিছু ডান্ডা, কিছু পেরেক, কিছু ইত্যাদি — সেই বদখত জটিল এলোমেলো চেহারাটাকে এমন নামময় করার রূপটান দেওয়াটাই অপারেটিং সিস্টেমের কাজ। শুধু সফট বা হার্ড ডিস্কের কঠিন চাকতিময় খুঁটিনাটি নয়, আমাদের ভুলতে দেয় এদের ইন্টারাপ্টগুলোকেও।

ইন্টারাপ্ট মানে কিছু সিগনাল যা বিভিন্ন ভৌত উপাদানগুলো থেকে অপারেটিং সিস্টেমের কাছে আসে, যেমন সিডি-ড্রাইভ, মাউস ইত্যাদি। অপারেটিং সিস্টেম এই ভৌত উপাদানগুলোকে মাথায় রাখে এই সিগনালদের দিয়ে। সিগনালগুলো আসে একটা ডিভাইস ড্রাইভার বা উপাদান চালক মারফত। ডিভাইস ড্রাইভার মানে বিশেষ রকমের কিছু আদেশ-সমাহার। যা দিয়ে অপারেটিং সিস্টেম একটা উপাদানকে ব্যবহার করে। ধরুন আপনি হার্ড-ডিস্ককে কাজে লাগানোর সময় যে সব আদেশ দিয়ে অপারেটিং কাজ করতে পারবে, মাউস বা সিডি বা স্ক্যানার ব্যবহার করার সময় সেই আদেশগুলো দিয়ে চলবে না। সমাহারটা বদলে যাবে। মাউসের জন্যে লাগে মাউস ড্রাইভার। আপনার মনিটরটাকে ব্যবহার করার জন্যে লাগে মনিটর ড্রাইভার। এই রকম আছে প্রত্যেকটা উপাদানেরই।

শুধু হার্ডওয়ার বা ডিভাইসদের ইন্টারাপ্টগুলোকে ভুলে থাকতে দেওয়া নয়, এই অপারেটিং সিস্টেমের কল্যাণেই আমরা ভুলে থাকতে পারি টাইমার-গুলোকে, কম্পিউটারের গভীর জটিল রহস্যময় উদরে এবং অন্ত্রে নিয়ত খেলাধূলামান ক্লক সাইকেলের ওঠানামাকে। ভুলে থাকতে পারি কতটা মেমরি বা স্মৃতিকে কী ভাবে ব্যবহার করা হচ্ছে — প্রতিমুহূর্তে — তার জটিল ছকটাকে।

ধরুন আপনি একটা কবিতা লিখছেন ওয়ার্ড-প্রসেসরে, তার ফাইলটা আছে ফ্লপিতে, আপনি যে বদলগুলো আনছেন সেগুলো লিখে রাখতে বলেছেন ফ্লপি নামক ভাঁড়ারে অমুক নামের ফাইলে। একটু আগে লেখা এবং হার্ড ডিস্কে সেভ করে রাখা আপনার যুগান্তকারী প্রোগ্রামগুলো কম্পাইল হচ্ছে অন্য আর এক দিকে। কবিতা লেখার প্রহরটাকে যে নান্দনিক উত্তাপে আপনি ভরে তুলছেন, সেই যে-নামেই-ডাকো-গোলাপের ছবিগুলো আছে একটা গ্রাফিক্স-

সিডিতে। সেখান থেকে একটা একটা করে গোলাপ তুলে এনে স্ক্রিনের ভারচুয়াল ফুলদানিতে ভরে দিচ্ছে একটা গ্রাফিক্স সফটওয়ার।

এবার, নতুন লাইনে তা দিতে গিয়ে আপনি সেই ছবির ফুল থেকে ফুলান্তরে গেলেন। আপনি তো কমান্ড দিয়ে বা মাউস টিপেই খালাস, কম্পিউটারকে তখন কী কী করে যেতে হচ্ছে নির্বাক বশংবদতায়, সেটা মাথায় এনেছেন কখনো? নিমেষে এই প্রতিটি নামের ঠিকানায় প্রতিটি চালু থাকা কাজের প্রতিটি কার্যরত ফাইলের সমস্ত একটা বিরাট অংশ তথ্যকে আপাতত অস্থায়ী একটা রকমে হার্ডডিস্কের সোয়াপ-ফাইল নামে একটা বিশেষ জায়গায় লিখে রেখে জ্যান্ত র‍্যামে তুলে আনতে হল ওই নতুন ফুলের নতুন কোটি কোটি পিক্সেলের নতুন অর্বুদ অর্বুদ বাইট সমগ্রকে — এই গোটা কাজটাই হয়ে গেল আপনার অগোচরে (হার্ড ডিস্কে অস্থায়ী রকমে লিখনীয় এই সোয়াপ ফাইল নামের জায়গাটার কথা পরে বারবার করে জানতে হবে আমাদের)। আপনি শুধু নাম ধরে ডেকেই খালাস, কাজের নাম ধরে কাজটা করার কথা জানিয়েই খালাস। আপনার হয়ে এই গোটাটাই মাথায় রেখেছে অপারেটিং সিস্টেম। আপনি যখন অমুক পার্টিশনের অমুক ডাইরেক্টরির অমুখ ফাইলে অমুক কাজ করতে বলছেন, এই প্রত্যেকটা অমুকই ভৌত নয় সাঙ্কেতিক, সে বুঝে নিচ্ছে সেই সাঙ্কেতিক অমুক মানে কোন ভৌত তমুক, জানে এবং হিশেব করে নিচ্ছে, নিঃশব্দে। এইভাবেই আপনার মেশিনকে মেশিনতর করে মেশিনোত্তরতায় ঠেলে দিচ্ছে, হার্ডওয়ার থেকে সফটওয়ারের জগতে টেনে আনছে সাঙ্কেতিক ভূমি মারফত — অপারেটিং সিস্টেম নামের ওই নির্বাক দাস।

৩।। ভৌত উপাদানদের নিয়ে তৈরি সাম্রাজ্য

কিন্তু অপারেটিং সিস্টেমকে ব্যবহারকারীর কাছে এই রকম একটা সহজ পারস্পরিকতা বা কনভিনিয়েন্ট ইন্টারফেস হিশেবে দেখাতে গিয়ে আমরা গোটা ব্যাপারটাকে উপর থেকে দেখছি — ব্যবহারকারীরা যেখানে রয়েছে — একটা মেশিনের হাই লেভেল বা উঁচু স্তর থেকে। ইউজার যেখানে প্রয়োগ করে চলেছে হাজার একটা অ্যাপ্লিকেশন। উল্টো একটা দৃষ্টিকোণ থেকেও পুরোটাকে দেখা যেতে পারত — হার্ডওয়ারের দিক থেকে, নিচুর দিকে থেকে, মানে গোটা কম্পিউটার সিস্টেমটাকে তার মাথার উপরে শীর্ষাসন করিয়ে। যখন আমাদের সামনে রয়েছে ওই ভৌত উপাদানের রসদগুলো তাদের ভৌত বাস্তবতা নিয়ে, তাদের ক্ষমতা এবং ক্ষমতার সীমা নিয়ে। এটাই কম্পিউটারের রসদের ভাঁড়ার, যাকে ব্যবহার করে কাজ করে চলতে পারে ওই অ্যাপ্লিকেশনগুলো। এই বিভিন্নমুখী বিচিত্র উপাদানের রসদগুলোকে সামাল দেওয়ার কাজটাও অপারেটিং সিস্টেমের। এর পরের সেকশনে আমরা ফেরত আসব রসদের ভাঁড়ারঘরের ভাণ্ডারীর ভূমিকায় অপারেটিং সিস্টেমের আলোচনায়। তার আগে এই রসদগুলোকে একটু দেখে নেওয়া যাক।

সচরাচর আমরা যে পিসিগুলোকে দেখি তাদের ক্ষেত্রে এই ভাঁড়ারের সবচেয়ে জনপ্রিয় উপাদানগুলো নিয়ে আমরা শূন্য নম্বর দিনে আলোচনা করেছি। কিন্তু সেখানে আমরা একটা পিসিকে বাইরে থেকে, একটা ব্যবহারকারীর চোখ থেকে দেখছিলাম। এখানে আমরা দেখছি এদের এক একটা রসদ হিশেবে, কার্যকারিতার আকর হিশেবে, অপারেটিং সিস্টেম তাদের এক এক জনকে দিয়ে এক একটা কাজ করিয়ে নেয়। অপারেটিং সিস্টেমের অন্তর্গত অংশ হিশেবে এদের এবার ভাবছি আমরা। এদেরকে ভিতরে নিয়েই তৈরি হয় একটা অপারেটিং সিস্টেম। অপারেটিং সিস্টেম — এই ব্যাপারটা একটা বিমূর্ত ধারণা, সেটা আকার পায়, মূর্ত হয়ে ওঠে এই উপাদানগুলোর শরীরেই। কাজ করতে থাকা সিপিইউটা একটু অপারেটিং সিস্টেম, হার্ডডিস্কটাও তাই, মেমরিও তাই। একটু আগে দেওয়া বাস-সংযোগের ছবিটা একবার দেখে নিন, সিপিইউ, মেমরি, আই-ও ডিভাইস, এরা সবাই একটা সিস্টেম বাস দিয়ে সংযুক্ত থাকে পরস্পরের সঙ্গে, তৈরি হয় একটা সমগ্র, যে সমগ্রটা হল অপারেটিং সিস্টেমের সাম্রাজ্য। শর্তহীন সম্রাট সে। আর সবাই সেখানে ছোট ছোট খণ্ডের মনসবদার জায়গিরদার, সম্রাটের সিদ্ধান্তের অধীন। আধুনিক একটা পিসিতে বাসের কাঠামো অবশ্য ওর চেয়ে অনেক জটিল হয়। একাধিক বাসের একটা সমাহার কাজ করে সেখানে। তবে সেসব জটিলতায় আমরা যাব না আমাদের এই পাঠমালায়।

> মাদারবোর্ড — নানা কোম্পানির নানা ক্ষমতার মাদারবোর্ড হয়, নানা ধরনের। এক একটা মাদারবোর্ডে এক এক ধরণের ভৌত উপাদান লাগানো যায়, এক এক মাত্রার এক এক মানের কাজ হয়। তাদের দামও হয় আলাদা আলাদা। কম্পিউটারের অন্য উপাদানগুলো কী ক্ষমতার কী মানের হবে তা মোটামুটি ভাবে নির্দিষ্ট

হয়ে যায় এই মাদারবোর্ড দিয়েই। মাদারবোর্ড গোটা ব্যাপারটাকে কোল পেতে দেয়। তার গায়েই লাগানো থাকে প্রসেসর, র‍্যাম, রম, আইও ডিভাইস। তার গায়ের সার্কিটগুলো দিয়েই কাজ করে বাস সংযোগ। বিদ্যুৎ সরবরাহকে প্রতিটি একক ভৌত উপাদানের কাছে পৌঁছে দেয় মাদারবোর্ড। কোনো একটা পিসিতে ঠিক কোন মানের কোন ক্ষমতার কী কী উপাদান লাগানো যাবে, প্রসেসর থেকে শুরু করে প্রতিটি কিছু, স্থির হয়ে যায় মাদারবোর্ড দিয়ে। সেই মাদারবোর্ডে কোন ধরনের কোন উপাদান ব্যবহার করা যাবে, এবং কোন বা কোন কোন উপাদান ব্যবহার করা যাবেনা, তার নির্দিষ্ট সব নিয়ম আছে। এই মাদারবোর্ডের ভৌত ব্যাকরণ তাই অনেকটাই স্থির করে দেয় অপারেটিং সিস্টেমের কাজের প্রকরণকে।

প্রসেসর — এই প্রসেসরই হল একটা কম্পিউটারের মস্তিষ্ক, আগেই বলেছি। বহু কোম্পানির বহু রকমের বহু গঠনের হতে পারে, গতি বা মেগাহার্জ হতে পারে অজস্র রকমের, কাজ করার একক মানে ওয়ার্ড হতে পারে ১৬ বিট থেকে ৬৪ বিট বা তারো বেশি এই নানা সাইজের, সবচেয়ে জনপ্রিয় ব্র্যান্ডগুলো অবশ্যই পেন্টিয়াম অ্যাথলন সাইরিক্স ইত্যাদি। এই প্রসেসরের কাজ হল মেমরি বা স্মৃতি থেকে নির্দেশ বা নির্দেশাবলী তুলে আনা এবং তাদের পালন করা। কোন ধরনের নির্দেশ কোন সিপিইউ পালন করতে পারবে তার খুব নির্দিষ্ট কিছু নিয়ম, এই রকম দু-একটা নিয়ম আমরা পরে উল্লেখ করব। সিপিইউ-র কাজ করার রকম, তার অংশগুলো, এবং তার রেজিস্টারগুলো সম্পর্কে একটা আবছা ধারণা আমাদের ইতিমধ্যেই হয়েছে। এর পরে এর কোনো কোনোটার সফটওয়ার অংশটা নিয়ে দু-চারটে কথা বলব আমরা। তবে এর চেয়ে বেশি কিছু আমাদের এই পাঠমালায় আসবে না। এটা এত বেশি টেকনিকাল এলাকা যে সেখানে কোনো কথা বলার যোগ্যতাই নেই আমার।

মেমরি — কম্পিউটারের ভৌত উপাদানগুলোর মধ্যে সিপিইউর পরেই গুরুত্বপূর্ণতম এই মেমরি। এর নানা ধরনের প্রকারভেদের কথা আমরা আগেই বলেছি, শূন্য নম্বর দিনে। র‍্যাম আর রম — এই দুইরকম স্মৃতি আমাদের সবচেয়ে বেশি পরিচিত। এর পরে আসে ক্যাশে, যা নিয়ে পরে আমরা অনেক খুঁটিয়ে আলোচনা করব। এর পরে আসে নানা ধরনের স্মৃতিভাঁড়ার, মানে মূলত ম্যাগনেটিক ডিস্ক মানে হার্ডডিস্ক বা ফ্লপিডিস্ক, আর অপটিকাল ডিস্ক মানে সিডি ডিভিডি ইত্যাদি। এদের নিয়ে বহু আলোচনা আমাদের এই পাঠমালায় বারংবার আসবে। আসবে হার্ডডিস্ক এবং সিডির আভ্যন্তরীন কাঠামো এবং সেখানের ফাইলব্যবস্থা নিয়ে দীর্ঘ আলোচনা। সিপিইউ রেজিস্টার, ক্যাশে, এবং র‍্যাম নিয়ে খুব ভালো করে জানব আমরা ভারচুয়াল মেমরি বা ভৌতিক স্মৃতির আলোচনায়, আট নম্বর দিনে গিয়ে। মেমরি এবং স্টোরেজ ডিভাইস — সবরকম স্মৃতিকেই মাপা হয় বাইট কিলোবাইট মেগাবাইট গিগাবাইট টেরাবাইটের মাপে, যাদের সম্পর্কেও পরে অনেক ভালো করে জানতে হবে আমাদের। হার্ডডিস্কের বেলায় আর একটা বড় জিনিষ হয়ে দাঁড়ায় তার আর-পি-এম বা রোটেশন পার মিনিট। মানে হার্ডডিস্কের মধ্যের প্ল্যাটারগুলো মিনিটে কতবার ঘোরে। ডিস্কটা কতটা দ্রুত কাজ করতে পারে সেটা এর উপরেই নির্ভর করে। প্রতিটি মেমরি উপাদানেরই অজস্র আলাদা আলাদা রকম আছে, যা আলোচনা করার জায়গা এটা নয়।

টাইমার — মাদার বোর্ডের যে সার্কিট্রি সময়গতি খেয়াল রাখে — তারিখ থেকে তারিখে এবং ক্রিয়া থেকে ক্রিয়ায়, মানে কোনো একটা কাজ কতক্ষণ ধরে চলছে — ইত্যাদি। এটা একটা গুরুত্বপূর্ণ উপাদান, অপারেটিং সিস্টেমের অনেকগুলো কাজ ব্যাপকভাবে নির্ভরশীল থাকে এই টাইমারের উপর। শুধু আমরা জানতে চাইলে সময় বলে দেওয়াটা নয়।

কিবোর্ড — নানা লে-আউটের, নানা গুণমানের, নানা রকম চাবি-সমাহারের। এ নিয়ে যথেষ্ট আলোচনা হয়েছে। এর পরে যেটা দরকার সেটা হল সেই কিবোর্ডটা প্রাকটিশ করা, গ্নু-লিনাক্স অপারেটিং সিস্টেমে কাজ করতে হলে টাইপটা আপনাকে এত প্রচুর করতে হবে, যে এটা খুব প্রয়োজনীয় ইনভেস্টমেন্ট। গ্নু-র তৈরি সফটওয়ার আছে, জিটাইপিস্ট, বেজায় চমৎকার জিনিষ।

পয়েন্টিং ডিভাইস — মানে মাউস (সিরিয়াল না পিএসটু না ইউএসবি বা হুইল বা অপ্টিকাল ইত্যাদি), বা ট্র্যাকবল বা এই জাতের অন্য কিছু, এটাতে কিছু আর জানানোর প্রয়োজন তো নেইই, বরং জানাটা একটু কমিয়ে ফেলতে পারলে ভালো বই মন্দ হবেনা, আরো যেহেতু আমরা আমাদের এই পাঠমালায় গুই বা গ্রাফিকাল-ইউজার-ইন্টারফেস যেহেতু ধরবই না।

কনসোল বা ভিডিও মনিটর — মনো প্রায় উঠেই গেছে এখন, কালার মনিটর অজস্র রকমের আয়তন এবং ক্ষমতার, তাদের পরস্পরের মধ্যে নাটকীয় তফাত, এই ক্ষমতার পার্থক্যকে আবার ব্যবহার করার মত সুযোগ থাকা চাই মাদারবোর্ডে সংযুক্ত ভিডিওকার্ড বা ভিডিও-কন্ট্রোলারের, যার মাধ্যমে কম্পিউটার কনসোলকে ভিডিও তথ্য পাঠায়। এক্স-উইনডোজ মানে গুই যেহেতু আমরা ছুঁচ্ছি না, তাই মনিটরের জটিলতা নিয়ে মাথা খারাপ করার মিনিমাম প্রয়োজন পড়বে না আমাদের।

প্রিন্টার — লেজার না ডেস্কজেট না ডটম্যাট্রিক্স — কত তার রেজলিউশন কটা পাতা প্রতিমিনিটে ছাপে ইত্যাদি। এটাও অপারেটিং সিস্টেমের একটা গুরুত্বপূর্ণ উপাদান।

নেটওয়ার্ক কলকব্জা — মোডেম বা আইএসডিএন বা ল্যান বা ওয়ান বা কত তার তথ্য স্থানান্তরের হার — কেবিপিএস বা এমবিপিএস ইত্যাদি। অন্য অনেক অপারেটিং সিস্টেমের চেয়ে গ্নু-লিনাক্স অপারেটিং সিস্টেমে এই নেটওয়ার্ক কলকব্জাটা আরো বেশি গুরুত্বপূর্ণ হয়ে ওঠে, তার কারণ তো আগেই বলেছি, গ্নু-লিনাক্স তথা যে কোনো ইউনিক্স সিস্টেমই প্রথমেই মেশিনটাকে নেটওয়ার্কিত বলে ধরে নেয়। হার্ডওয়ার নিয়ে তেমন কথা না-এলেও সফটওয়ার কনফিগারেশন নিয়ে কিছু আলোচনা আসবে আমাদের। নয় নম্বর দিনে গিয়ে।

এবং এগুলো ছাড়াও আরো অজস্র প্রচুর রকমের উপাদান এবং তাদের নিজস্ব নিজস্ব জটিলতা। এবং প্রায় প্রতিমুহূর্তে এদের নতুন নতুন ধরন বাজারে আসছে নতুন নতুন প্রকৌশল নিয়ে, নতুন নতুন কাজের প্রয়োজন মেটাতে। তারা সঙ্গে সঙ্গে তাদের ড্রাইভার সহ ঢুকে আসছে কারনেলে, অপারেটিং সিস্টেমে। এই ড্রাইভার ব্যাপারটা নিয়ে আলোচনা আজই হবে।

বারবার আমরা নানা কাজে বিট বাইট বা কিলোবাইট (কেবি) মেগাবাইট (এমবি) গিগাবাইট (জিবি) — এগুলো ব্যবহার করছি। বিট বা বাইট কাকে বলে আমরা জানি। এর পরে এককগুলো আমাদের অভ্যস্ত দশের গুণিতক থেকে একটু আলাদা। আমরা মেট্রিক সিস্টেমে অভ্যস্ত হই এক কিলো মানে এক হাজার। এখানে আসে হাজারের সবচেয়ে কাছের ২-এর গুণিতক। মানে ১০২৪, মানে ২ এর ১০-তম পাওয়ার। ১০২৪ বিটে এক কিলোবাইট বা কেবি। ১০২৪ কিলোবাইটে এক মেমাবাইট বা এমবি। ১০২৪ মেগাবাইটে এক গিগাবাইট বা জিবি। এর উপরে আছে আরো বড় বড় একক — টেরাবাইট পেটাবাইট এক্সাবাইট জিটাবাইট ইয়োটাবাইট ইত্যাদি। তালিকাসহ এদের নিয়ে বড় করে আলোচনা আসবে আট নম্বর দিনে। আপাতত এটুকুতেই কাজ চলে যাবে। আবার, মোডেম বা আইএসডিএন জাতীয় নেটওয়ার্ক উপাদানগুলোকে আমরা মাপি কত বাইট সে এক সেকেন্ডে স্থানান্তর করতে পারে তার নিরিখে। কেবিপিএস মানে কিলোবাইট-পার-সেকেন্ড। এমবিপিএস মানে মেগাবাইট-পার-সেকেন্ড। একটা কম্পিউটার প্রসেসর কত তাড়াতাড়ি কতটা কাজ করতে পারে তাকে মাপি একই ভাবে কিলো-হার্জ মেগা-হার্জ বা গিগা-হার্জ দিয়ে। এখানেও কিলো মানে ১০২৪। ১০২৪ কিলোতে এক মেগা, ১০২৪ মেগাতে এক গিগা। যাকগে, এবার ফিরে আসা যাক রসদের এই ভাঁড়ারঘরের ম্যানেজার হিশেবে অপারেটিং সিস্টেমের কথায়।

৪।। অপারেটিং সিস্টেম মানে রসদের ভাঁড়ারঘরের দায়িত্ব

এই ভাঁড়ারঘরের ম্যানেজার হিশেবে অপারেটিং সিস্টেমের কাজ হল এই নানামুখী নানাধাঁচের নানা রসদকে নিয়ন্ত্রিত এবং সংগঠিত করে হাজির করা বিভিন্ন সফটওয়ারের কাছে। আগের সেকশনের উদাহরণটাই ধরুন। একটা ওয়ার্ড-প্রসেসর, একটা কম্পাইলার, এবং একটা গ্রাফিক্স সফটওয়ার। আর এদের কাজ করার জন্যে নিচের স্তরে অনুচ্চারিত রকমে আরো অনেকগুলো সফটওয়ার, যে সিস্টেম সফটওয়ার গুলো একটু আধটু পরে আমাদের কাজে আসবে। এর মধ্যে একটা মূল রকম হল যখ বা ডিমন। যারা সিস্টেমের মধ্যে বসে থাকে এবং দিবারাত্র তার নিজের কাজ, মানে সিস্টেমে কী কী ঘটছে তার পাহারা দিয়ে চলে। পাহারা দেয় আর খাপ পেতে অপেক্ষা করে, ঠিক কখন সেই ঘটনাটা ঘটবে যখন তার কিছু একটা করতে হবে। কিছু একটা করা মানে মূলত অপারেটিং সিস্টেমের কাছে জানিয়ে দেওয়া, বস, দিস নিডস অ্যাকশন। যেমন ধরুন ক্রন ডিমন, তাকে মা-কালী বলে ডাকাই যায়, কালক্রম-অনুসরণ বা কাল-ই তার কাজ (কার্টসি রাঘব বন্দোপাধ্যায়), সময়ের টাইমার সার্কিটের দিকে তাকিয়ে তাকিয়ে তার আঁখি না-ফিরে। গ্নু-লিনাক্স অপারেটিং সিস্টেমে এরকম অনেক কাজ আছে যা সময় মেপে করতে হয়। চাইলে আপনি নিজেও কাজ দিতে পারেন কালী যখকে। আপনি অপারেটিং সিস্টেম নন বলে যে আপনার কাজটা রিফিউজ করবে, এরকম নীচ

মনই নয় তার। এবার ধরুন, আপনি বলে দিয়েছেন, ঠিক এই সময়ে এই কাজটা দরকার আপনার। সেই সময়টা আসা মাত্রই কালী যখ জানিয়ে দেবে অপারেটিং সিস্টেমকে, কাজের সময় হল।

এরকম অনেক যখ আছে আপনার সিস্টেমে, আরো অনেক কিছু আছে। আপনি তাদের কথা জানুন আর না-জানুন, আপনার মেশিনে কার্যরত জ্যান্ত ব্যবস্থাটার গভীর গোপন অন্দরে তারা ছেদহীন যতিহীন ত্রুটিহীন ভাবে কাজ করে চলেছে। পরে আমরা দেখব, এরা যে আছে, সেই উপস্থিতিটাকে ফুটিয়ে তোলার দেখার নানা উপায়। যেমন একটা হল, মেশিনের মধ্যে একটা কোনো মুহূর্তে চলমান প্রতিটি প্রক্রিয়ার একটা তালিকা নেওয়া। গ্নু-লিনাক্স সিস্টেমে তার জন্যে কমান্ড বা আদেশ দেওয়া যায়, ‘`ps aux`’। এরকম যে কোনো কমান্ডই কমান্ড প্রম্পটে দেওয়ার পরে এন্টার মারলে কম্পিউটার সেই আদেশটা পালন করে। এই কমান্ডটুকুর মধ্যে ‘ps’ অংশটা হল মূল আদেশ, আর ‘aux’ তার অপশান বা বিশেষ নিয়ন্তা। এগুলো আমরা পরে বুঝব, কমান্ড, কমান্ডের গঠন, তার অপশন ইত্যাদি। ‘ps’ কমান্ডটা স্ক্রিনে একটা তালিকা দেখায় চলমান প্রক্রিয়াগুলোর বা প্রসেসের। পরে আমরা দেখব এই এক একটা কাজকে ডাকা হয় এক একটা প্রসেস বলে। আমরা আগেই বলেছি প্রতিটি প্রোগ্রামই এক একটা ফাইল। একটা বিশেষ ধরনের ফাইল যারা নিজেরাই চলে। সক্রিয় বা এক্সিকিউটেবল বা বাইনারি ফাইল। এই ধরনের একটি বিশেষ ফাইলের একবার চলাকেই বলে একটা প্রোগ্রাম রান করা। একটি বিশেষ প্রোগ্রামের কোনো একবারের ক্রিয়াশীলতাকে বা রান-করা-কে আমরা ডাকি একটা প্রসেস বা প্রক্রিয়া বলে। একই সফটওয়ার প্রোগ্রামকে যদি দুবার আলাদা করে চালানো হয়, সেটাকে ধরা হবে দুটো প্রক্রিয়া। কোনো একটা প্রোগ্রাম যখন নিষ্ক্রিয় বা স্থির রয়েছে তখন সেটা ফাইল, যখন সে ক্রিয়াশীল, রান করছে, সেটা একটা প্রসেস বা প্রক্রিয়া। এই প্রসেস বা প্রক্রিয়াকে পরে আমাদের আরো সক্রিয় ভাবে বুঝবে — তার কথায় আমরা পরে আসছি।

প্রসেস বা প্রক্রিয়া বলতে এক কথায় ধরে নিন, একটা চলন্ত প্রোগ্রাম। ডিমন বা যখও একটা প্রসেস, আবার আপনি একটা গান বাজানোর অ্যাপ্লিকেশন চালাচ্ছেন, সেটাও প্রসেস, ওয়ার্ড প্রসেসর খুলে লিখছেন, বা জিসিসি দিয়ে কম্পাইল করছেন — এরা প্রত্যেকেই এক একটা প্রসেস। এই প্রসেসরা প্রত্যেকেই ব্যবহার করছে সিপিইউ বা প্রসেসরকে, মেমরিকে এবং কনসোল ইত্যাদি নানা ধরনের আইও মাধ্যমকে। অথচ রসদ তো অনন্ত না, একটা কম্পিউটারের মোট প্রসেসর সামর্থ কী, র‍্যাম কতটা, হার্ডডিস্ক কী সাইজের, প্রিন্টার কটা, কনসোল কী মানের — এরকম সবকিছুই তো প্রদত্ত, সেই মুহূর্তে অপরিবর্তনীয়। তাহলে এই সীমিত সামর্থকে ব্যবহার করেই কাজ করতে হবে প্রতিটি প্রক্রিয়ার। তাই, এদের মধ্যে নিয়তই একটা প্রতিযোগিতা চলছে — কোন প্রসেস কোন রিসোর্সকে কতটা সময়ের জন্যে ব্যবহার করতে পারবে। প্রসেসর, মেমরি, আইও — এই বিবিধ রসদকে একটা নিয়ন্ত্রিত এবং সংগঠিত রকমে বিভিন্ন চলমান সফটওয়ারের কাছে, মানে প্রসেসের কাছে, পৌঁছে দেওয়াই অপারেটিং সিস্টেমের কাজ। নিয়ন্ত্রিত এবং সংগঠিত রকমে। যাতে সব কটা অ্যাপ্লিকেশন তথা প্রক্রিয়াই তাদের নিজের নিজের কাজ করে চলতে পারে, তাদের গুঁতোগুঁতি না-করতে হয়, আবার সব দুধ শুধু ছাগলের প্রথম দুটো ছানাই খেয়ে গেল, তৃতীয়টা শুধু নেচেকুঁদে বেড়াল, এরকমটাও যাতে না-হয়।

ধরা যাক, তিনটে সফটওয়ারই তাদের কাজ একইসঙ্গে মনিটরে বা পর্দায় পৌঁছে দিতে চাইছে। কী দাঁড়াবে বিষয়টা? একটু সাইফি মানে সায়েন্স ফিকশন করা যাক। পর্দা জুড়ে, ধীরে, ঘুমন্ত বাচ্চার চোখ খোলার মত করে, ছোট ছোট কুচি কুচি যেগুলো ছড়িয়ে পড়ল প্রস্ফুটিত গোলাপ-পাঁপড়ির লাস্যময় শরীরে — না, সেগুলো কিন্তু পরাগ না, আপনার কবিতার যতিচিহ্ন — কোলন, সেমিকোলন, কমা। আর তিরতির করে নড়তে নড়তে এপাশ থেকে ওপাশ চলে যাচ্ছে, উদাসীন, ঋজুরেখ — সে বা তারা কোনো মধুলোভী মধুকরও নয়, নিতান্তই কম্পাইলেশনকালীন কম্পাইলারবার্তা। কবিতা, গোলাপ আর কম্পাইলেশন এখন জমাট দ্রবীভূত এক। মানে? মানে কেলেঙ্কারি। ক্যাওস। সাড়ে-বত্রিশ-ভাজা। তিনটে কাজকে তিনটে বাফারে বা অস্থায়ী মেমরিভূমিতে সংহত করে এটাকে একটা সংবদ্ধ বিস্রস্ত গোছালো জায়গায় নিয়ে আসাটা আশু প্রয়োজন, আর ঠিক সেই কাজটাই করে অপারেটিং সিস্টেম।

যখনি একটা কাজ শেষ হল, ধরুন কবিতাটা, তারা স্থান পেয়ে গেল আপনার কবিতাদের জন্য নির্দিষ্ট কোনো বিশেষ নামের ডিরেক্টরিতে, সেই এক বা একাধিক কবিতার জন্য ন্যস্ত একটি বিশেষ নামের ফাইলে। এবং এই পুরো সময়টা জুড়ে অন্য কাজগুলোকে কোনো ধরনের কোনো বিরক্তিই কিন্তু করা হচ্ছে না। ভাই, আমার সিস্টেমের রসদ দিয়ে আমি এখন কবিতার ফাইল লিখছি, তোমরা অন্য অন্য ছাগশিশুগন একটু দৈত্যের বাগানে পর্যটন করে এসো,

এরকমটা কিন্তু বলে দিচ্ছেনা অপারেটিং সিস্টেম। সবাই, প্রত্যেকেই ঘটে চলেছে, প্রতিমুহূর্তেই, একবারো না-থেমে। এই লিভ অ্যান্ড লেট লিভের গনতন্ত্রটাই ঘটছে অপারেটিং সিস্টেমের দৌলতে।

অনেকগুলো সমান্তরাল ভাবে চলমান প্রক্রিয়ার এই জটিল কেচ্ছাটা আরো কেলোপরায়ণ হয়ে ওঠে যখন সিস্টেমটা একটা একটা বহু-ব্যবহারকারী বা মাল্টিপল-ইউজার সিস্টেম। আর আগেই তো বলেছি, যে কোনো গ্নু-লিনাক্স বা ইউনিক্স সিস্টেমই আভ্যন্তরীণ ভাবে সেরকম করেই তৈরি। ধরুন এগারো জন ব্যবহারকারী এগারোটা টার্মিনাল থেকে একত্রে লগ-ইন করেছেন সিস্টেমে, একই সঙ্গে লিখছেন এগারো রকমের টেক্সট, কবিতা থেকে হালখাতা। এবং এই টেক্সট লিখে-চলা-কালীন তারা চালাচ্ছেন তিনটে করে প্রোগ্রাম, মানে এগারো গুণ তিন তেত্রিশটা সফটওয়ার চালাচ্ছেন মোট। তেত্রিশখানা জীবন্ত ধাবমান প্রসেস। এর সঙ্গে আছে গোদের উপর বিষফোঁড়ার মত অগুন্তি সিস্টেম প্রসেস, সিস্টেমের কাজ করে চলার জন্যেই অত্যাবশ্যক কিছু প্রক্রিয়া, ওই ডিমন বা যখদের মত। আর এই এগারো জনের ইউজার টিম, এরা প্রত্যেকেই তাদের রচনাকে কম্পিউটার মেমরিতে ধৃত নিরালম্ব বায়ুভূত ইলেকট্রনিক থেকে বাস্তব কাগজের আর কালির প্রিন্টের আকারে নিয়ে আসছেন একমেবাদ্বীতিয়ম নেটওয়ার্কাবদ্ধ প্রিন্টার থেকে। এরকম একটা অবস্থায় এই টিমের একজন হিশেবে আপনি যেই নিজের কবিতার প্রিন্টআউট চাইলেন, সেই শুদ্ধ ভুর্জপত্রে আপনার কোনো গোলাপোল্লাসের পরের লাইনেই যে, ‘গরুর খোল উনিশ মন’, বা জিসিসির কোনো বার্তা ছাপা হচ্ছে না, তার একমাত্র কারণই ওই অপারেটিং সিস্টেম। অপারেটিং সিস্টেমই পৌঁছে দিচ্ছে গরুকে তার গোয়ালে, চারি ভর্তি খোলকে গরুর চোয়ালে, এবং গোলাপকে ফুলদানিতে।

বহু ব্যবহারকারী। বহু ফাইল তাদের সম্পূর্ণ নিজস্ব। বহু ফাইল সর্বসাধারনের। বহু ফাইল আবার একমাত্র রুট বা ওই সিস্টেমের একচ্ছত্র শাসকের। ইউনিক্স বা যে কোনো গ্নু-লিনাক্স সিস্টেমে এরকমই নিয়ম, উইন্ডোজ এক্সপি-র বেলাতেও অনেকাংশে যা সত্যি। এই অজস্র ধরনের মালিকানার ফাইলের উপর অজস্র ধরনের অধিকার, সেগুলোর উপর একই সঙ্গে কাজ করছে বহু ব্যবহারকারীর চালু করা বহু প্রক্রিয়া, এরা প্রত্যেকেই দৌড়চ্ছে মানে রান করছে একই সঙ্গে। এর সঙ্গে রয়েছে অপারেটিং সিস্টেমের নিজের শুরু করা নিজের জন্যেই প্রয়োজনীয় অনেকগুলো প্রক্রিয়া। অপারেটিং সিস্টেমকে তাই প্রতিমুহূর্তে হিশেব করতে হচ্ছে কোনো একটা বিশেষ সময়বিন্দুতে কোন রসদকে ব্যবহার করছে কোন ব্যবহারকারীর কোন প্রক্রিয়া। সে রয়েছে যাবতীয় রসদের ভাঁড়ারের দায়িত্বে।

৫।। সমান্তরাল বহুক্রিয়া বা মাল্টিপ্লেক্সিং

পরে আমরা দেখব নির্দিষ্ট পরিমাণ রসদকে এই বিবিধ প্রক্রিয়ার মধ্যে ভেঙে দেওয়ার জন্যে অপারেটিং সিস্টেমের একটা নিজস্ব পদ্ধতি আছে। চলমান সমস্ত প্রক্রিয়া বা প্রসেসের মধ্যে সীমিত রসদ শেয়ারিং বা ভাগবাটোয়ারার যে পদ্ধতির টেকনিকাল নাম মাল্টিপ্লেক্সিং। দুটো আলাদা আলাদা তলে এই বণ্টনটা ঘটে। একটা তল হল ভূমি বা স্পেস। অন্য তলটা কাল বা টাইম।

মাল্টিপ্লেক্সিং বা বহুক্রিয়তা (জঘণ্য শোনাচ্ছে, কিন্তু এর চেয়ে ভালো কিছু মাথায় আসছে না প্রতিশব্দ হিশেবে, এর পর থেকে মাল্টিপ্লেক্সিং-ই লিখব) গোটা ব্যাপারটা অত্যন্ত জটিল। আমি লিখব কী, আমার নিজেরই আন্দাজ অত্যন্ত ভাসাভাসা, যতটুকু না-জানলে গ্নু-লিনাক্সের মাল্টি-টাস্কিং বা একই সঙ্গে একাধিক কাজ করতে পারার বাড়তি সক্ষমতাটাকে ঠিক ভাবে বুঝে ওঠা যায়না। আসলে আমার নিজের গ্নু-লিনাক্স অভিজ্ঞতার প্রথম বিস্ময়গুলোর একটা ছিল এটাই। তার আগেকার অনেকগুলো বছরের ডস-উইন্ডোজ করতে গিয়ে একভাবে অভ্যস্ত ছিলাম। সিনেমায় বা তথ্য চিত্রে যখন দেখতাম একই সঙ্গে কতগুলো করে উইন্ডোজ খোলা, কিম্বা, অনেকগুলো করে কাজ একই সঙ্গে হচ্ছে, ভাবতাম ওগুলো সিনেমার ম্যাজিক, সেই লাস্ট অ্যাকশন হিরো সিনেমার বাচ্চাটা শোয়ার্জনেগারকে যেমন বলেছিল, ‘দেখে বুঝতে পারছ না, এটা বাস্তব জীবন নয় সিনেমা — দেখছ না চারদিকের প্রত্যেকটা মেয়েই কেমন এলিগ্যান্ট’। বা ভাবতাম, প্রথম বিশ্বের প্রথম শ্রেণীর হার্ডওয়ারের ক্যালি। গ্নু-লিনাক্স করার প্রথম দিকে সিস্টেমের মধ্যে দেওয়া হাউ-টু পড়ে যখন দেখতাম, অনেক ভালো মাল্টিটাস্কিং-এর কথা বারবার উল্লিখিত, তখনো বুঝিনি, ব্যাপারটা আসলে কতদূর সত্যি। এখন তো অভ্যস্ত হয়েই গেছি, একই সঙ্গে অনেকগুলো কাজ চালিয়ে, কিন্তু সেটা উইন্ডোজ অপারেটিং সিস্টেমের নিরিখে এতটাই অবিশ্বাস্য যে আমি আলাদা করে বলতে যাচ্ছি না, নিজেই করে দেখুন। আমার মেশিনের হার্ডওয়ার খুব একটা অগ্রবাহিনী নয় আদৌ, অ্যাথলন ১৭০০ প্রসেসর, এএন২৬৬ভিএম

মাদারবোর্ডে, ২৫৬ এমবি র‍্যাম, আশি জিবি হার্ডডিস্ক, এবং দুটো সিডি ড্রাইভ, একটা রাইটার একটা রিডার, একটা ডটম্যাট্রিক্স প্রিন্টার। তাতেই এতকিছু করা যায়। একটু ভালো হার্ডওয়ার লাগালে মনে হয় একসঙ্গে চারপাঁচটা বই লেখা যেত।

যখন একটা বিশেস রসদকে সময় বা টাইম-এর নিরিখে মাল্টিপ্লেক্স করা হয়, বিভিন্ন আলাদা আলাদা প্রক্রিয়া তখন খাপ পেতে থাকে, কখন আগের প্রক্রিয়াটা রসদটাকে ছেড়ে দেবে, এবং সে রিসোর্সটার উপর নিজের অধিকার কায়েম করবে। প্রতি সেকেন্ড সময়কে অনেকগুলো ছোট ছোট টুকরোয় ভেঙে ফেলা হয়। আগেই বলেছিলাম, আপনি আপনার মেশিনের অপারেটিং সিস্টেমের কাছে ব্রহ্মা বললেই হয়, আপনার চোখের পাতা একবার পড়তে পড়তে তার হাজার যুগ চলে যায়। সময়ের এই অজস্র ছোট ছোট টুকরোর এক একটা টুকরো জুড়ে একটা বিশেষ রসদ থাকে একটা বিশেষ প্রক্রিয়ার দখলে। তার পরের সময়ের টুকরোয় ওই রসদটা চলে যায় পরের প্রক্রিয়ার হাতে। প্রথম প্রক্রিয়ার পরে পায় দ্বিতীয়, তারপরে তৃতীয়, এরকম ঘুরতে থাকে। শেষ প্রক্রিয়াটার পরে আবার ঘুরে আসে প্রথম প্রক্রিয়া।

এটাকে মাল্টি-টাস্কিং বলেও ডাকা হয় — তবে সেটা এই অর্থে যে একই সঙ্গে কম্পিউটার একাধিক টাস্ক-এ বা কাজে রত থাকছে। যেমন ধরা যাক, একটি এক-প্রসেসর কম্পিউটার সিস্টেমে (একটা কম্পিউটারে প্রসেসরের সংখ্যা একের বেশি হতেই পারে, যেমন সিমেট্রিক-মাল্টি-প্রসেসিং বা এসএমপি, যদিও সচরাচর, বিশেষত পিসি বা পার্সোনাল কম্পিউটারের বেলায় আমরা একটা প্রসেসরের মেশিনেই অভ্যস্ত) প্রথম সময়-টুকরোটা পাচ্ছে প্রথম প্রক্রিয়া — সেই সময়টুকু জুড়ে প্রথম প্রক্রিয়া ব্যবহার করে নিচ্ছে প্রসেসরটাকে। পরের সময়-টুকরোয় প্রসেসর চলে যাচ্ছে দ্বিতীয় প্রক্রিয়ার দখলে। এই ভাবে ঘুরছে। কোন প্রক্রিয়া কত সময় ধরে প্রসেসরকে পাবে, যার পর তাকে ছেড়ে দিতে হবে — এটা ঠিক করে দেওয়া-ই অপারেটিং সিস্টেমের কাজ। প্রিন্টারও কাজ করে অনেকটা এই ভাবেই। যখন অনেকগুলো প্রিন্ট জব বা ছাপার কাজ লাইনে আছে — সকলেই অপেক্ষা করছে প্রিন্ট হবে প্রিন্ট হবে বলে, প্রিন্টারের স্পুলদেহে আসিতেছে চলে, প্রিন্ট হয়ে আসিয়াছে যারা, প্রিন্টিত হতে হয় যাহাদের, এবং এতগুলো শবরী-প্রিন্টজবের পাথর ছুঁয়ে দেওয়ার জন্যে রাম মাত্র একজন, মানে প্রিন্টারের সংখ্যা এক — তখন অপারেটিং সিস্টেমই সিদ্ধান্ত নেয় কোন প্রিন্ট কার পরে এবং কখন শুরু হবে, কার পরে কে আসবে।

অন্য ধরনের মাল্টিপ্লেক্সিং-টা হল ভূমি বা স্পেস ভিত্তিক মাল্টিপ্লেক্সিং। একটা লাইনে দাঁড়ানো খদ্দেররা একের পর এক আসছে — একজনের হয়ে গেলে পরের জন, এবং প্রতিটি খদ্দেরেরই চাহিদা অশেষ, তারা একটা মাল পাওয়া মাত্রই ঘুরে গিয়ে আবার লাইনে দাঁড়িয়ে পড়ছে পরের মালের জন্যে — কালভিত্তিক মাল্টিপ্লেক্সিং-এ এই ভাবে কাজ হচ্ছিল। এবার পদ্ধতিটা আর সময়ের নিরিখে হচ্ছে না, হচ্ছে ভূমির নিরিখে। মানে রসদটাকে একটা জমি বলে ভাবুন, মেশিনের প্রাপ্তব্য মোট জমি। এবং যখনই কোনো প্রক্রিয়া তার ঘর বাঁধতে চাইছে তখনই ওএস মানে অপারেটিং সিস্টেম (OS — **O**perating-**S**ystem) তাকে সেই মোট জমির থেকে একটুকরো জমি বরাদ্দ করছে, এখন থেকে এই জমিটুকু তোমার। এই ঘর-বাঁধাটা কখনো টেম্পোরারি কখনো পার্মানেন্ট। কোনো প্রক্রিয়া হয়তো কয়েক লহমার একটা টেম্প বা অস্থায়ী ফাইল লিখেই খুশি, তার কাজ শেষ হলেই উবে যাচ্ছে কাগজের ঘর, জমি ফেরত চলে যাচ্ছে ওএস-এর রাজকোষে। আর কোনো প্রক্রিয়া হয়তো বের হয়ে আসার আগে তার সমস্ত ফলাফল ছেপে দিচ্ছে কাগজের পাতার প্রিন্টে বা সিডির অপটিকাল চাকার পাতায় লেজারের আখরে।

এই ভূমি ভিত্তিক মাল্টিপ্লেক্সিং ব্যবস্থায় প্রতিটি খদ্দেরই একটা রিসোর্সের এক একটা টুকরোকে পেয়ে যাচ্ছে। সময়কে টুকরো না করে এখানে টুকরো করা হচ্ছে রসদটাকে। যেমন মূল র‍্যান্ডম-অ্যাকসেস-মেমরি বা র‍্যামকে ভাঙা হয় এই ভাবেই। একটা সময় বা কালভিত্তিক মাল্টিপ্লেক্সিং পদ্ধতির কথা ভাবুন। পদ্ধতিটা চলছে, মানে, একই সঙ্গে জ্যান্ত একাধিক প্রক্রিয়া চলছে মেশিনে। এবং একটু বাদে বাদে এক এক টুকরো সময় জুড়ে সিপিইউকে ব্যবহারের অধিকার এক একটা প্রোগ্রামের হাতে ছেড়ে দিচ্ছে ওএস। এবার ভাবুন, এটা যদি চালিয়ে যেতে হয়, তাহলে এই কালভিত্তিক মাল্টিপ্লেক্সিং-এর স্বার্থেই একটা ভূমিভিত্তিক মাল্টিপ্লেক্সিং পদ্ধতি-ও চালাতে হবে ওএস-কে। একাধিক ক্রিয়াশীল প্রোগ্রাম বা প্রসেসের জন্যে মোট মেমরিটাকে টুকরোয় টুকরোয় ভেঙে ফেলতে হবে। যাতে একই সঙ্গে একাধিক প্রোগ্রাম র‍্যামে থাকতে পারে। কারণ একটু বাদে বাদেই তো প্রত্যেকের-ই পালা আসছে প্রসেসর বা সিপিইউ ব্যবহার করার, কালভিত্তিক মাল্টিপ্লেক্সিং-এর নিয়ম মেনে। আর কোনো একটা প্রোগ্রাম যদি সিপিইউ-কে

ব্যবহার করতে চায়, ওই প্রোগ্রাম চলার জন্যে প্রয়োজনীয় গোটা তথ্যটাকেই তো মেমরিতে থাকতে হবে — যাতে চাওয়া মাত্র সিপিইউ তাকে মেমরিতে খুঁজে পেতে পারে। তাই সবকটা প্রোগ্রামকেই একই সঙ্গে মেমরির মধ্যেই বসবাস করতে হবে, অল্প অল্প জমি নিয়ে, তেঁতুল পাতায় ন-জনের মত। একবার ভেবে দেখুন, বাড়তি ভৌতিক স্মৃতি বা ভারচুয়াল মেমরির প্রয়োজন পড়ে এখান থেকেই। সীমিত র‍্যামে যখন গোটাটা আর রাখা যাচ্ছেনা, ওএস তখন আর একটা বাড়তি জমিকে অস্থায়ী রকমে র‍্যামের প্যালা হিশেবে নিয়ে আসছে। এবার র‍্যামে তাদের কানটা রেখে দেওয়া হচ্ছে, যাতে প্রয়োজন পড়লেই সিপিইউ র‍্যামে থাকা তাদের কান ধরে টেনে-ই ভারচুয়াল মেমরি বা ভৌতিক স্মৃতি থেকে তাদের মাথাটা পেয়ে যেতে পারে।

অন্য আর এক ভাবেও এই গোটাটা করা যেতে পারত। তা হল গোটা মেমরিটাই একটা প্রোগ্রামকে দিয়ে দেওয়া, তারপর, তার পালা শেষ হলে আবার গোটা মেমরিটাকেই পরের প্রসেস বা প্রক্রিয়ার জন্যে ছেড়ে দেওয়া। বেশিরভাগ ক্ষেত্রেই, যদি পর্যাপ্ত র‍্যাম থাকে, এই দ্বিতীয় পদ্ধতিটা অর্থহীন। কারণ, স্বাভাবিক একটা ওএস-এ, স্বাভাবিক অবস্থায়, বেশির ভাগ প্রোগ্রামেরই সাইজ যা তাতে গোটা র‍্যামের কোনো প্রয়োজনই পড়ে না। গোটা র‍্যামের একটা ছোট্ট ভগ্নাংশতেই তারা ধরে যায়। আর একটা রসদ যাকেও এই একই ভাবে ভূমি বা স্পেসগত মাল্টিপ্লেক্সিং করা হয় — সেটা হল হার্ডডিস্ক। একই সঙ্গে একই হার্ডডিস্কের শরীরে খচিত থাকে অজস্র প্রোগ্রাম ফাইল এবং তাদের কাজ করার জন্যে প্রয়োজনীয় আরো প্রচুর ফাইল। একই সঙ্গে ক্রিয়াশীল একাধিক প্রসেস তাদের কাজের ফাইলগুলো লিখতে এবং পড়তে থাকে একই হার্ডডিস্কে। এবং গ্নু-লিনাক্স বা ইউনিক্স এর বেলায় এর এক একটা প্রসেস হয়ত চালাচ্ছে এক একজন ইউজার। হার্ডডিস্কের কোন অংশটা কখন কোন প্রসেসের জন্যে বর্তাবে সেটা ঠিক করে দেয় এই ওএস বা অপারেটিং সিস্টেম। কী ভাবে করে সেটা আমরা জানব ছয় এবং সাত নম্বর দিনে ফাইলসিস্টেম জানার সময়।

৬।। আইও ডিভাইস, ডিভাইস কন্ট্রোলার এবং ইন্টারাপ্ট

আগের সেকশনটা দেখুন, আমরা রসদের উদাহরণ দিতে গিয়ে বারবার নানা ধরনের স্মৃতির কথাই বলেছি মূলত। মেশিনের মূল স্মৃতি র‍্যাম, বা হার্ডডিস্ক জাতীয় নানা ধরনের স্মৃতির ভাঁড়ার বা স্টোরেজ ডিভাইস। কিন্তু রসদ মানে কিন্তু আদৌ শুধু স্মৃতি নয়। এর মধ্যে পড়ে নানা ধরনের আইও বা ইনপুট-আউটপুট ডিভাইস। যার বেশ কিছু উদাহরণ ইতিমধ্যেই দিয়েছি আমরা, আজ এবং শূন্য নম্বর দিনে। অপারেটিং সিস্টেমের এই রসদ-বন্টন রাজত্বের খাস প্রজা অবশ্যই মেমরি। তার প্রসঙ্গে সামান্য কিছু কথা আমরা আগেই বলেছি, এক নম্বর দিনের গোড়ায়। আরো পরে আসব।

আজকের আলোচনার ২ নম্বর সেকশনের ছবিটা একবার দেখে নিন। যেখানে আমরা মেমরি এবং প্রসেসরের সঙ্গে পেরিফেরাল উপাদানগুলোর বাস-সংযোগের ছকটা দেখিয়েছিলাম। আমরা এখন সামান্য কিছু কথা বলে নেব এই আইও উপাদান এবং বাস নিয়ে। মেমরির পরেই অপারেটিং সিস্টেমের সবচেয়ে বড় ব্যস্ততাটা থাকে এই ইনপুট-আউটপুট নিয়ে। বাস-সংযোগের ছবিটায় দেখুন, প্রতিটি আইও উপাদানের মূলত দুটি অংশ — একটা হল মূল উপাদান বা তার ভৌত শরীর, এবং অন্য অংশটা হল তার নিয়ন্ত্রণ। কনসোল, কিবোর্ড, ফ্লপি-ডিস্ক এবং হার্ডডিস্ক — এই চারটেকে দেখিয়েছি আমরা ছবিতে, এরকম আরো হতে পারে। ধরুন, ফ্লপি ডিভাইস বলতে আমরা বুঝি ফ্লপি ঢোকানোর এবং চালানোর ড্রাইভটা, এই ভৌত যন্ত্রাংশটা। আর তার নিয়ন্ত্রণটা ঘটে ফ্লপি-ডিস্ক কন্ট্রোলার বলে একটা জিনিষ দিয়ে। এই কন্ট্রোলার হল একটা চিপ সমেত সার্কিট। বা অনেক ক্ষেত্রে কয়েকটি চিপের মিলিত একটা কার্ড বা প্লাগ-ইন-বোর্ড, যাদের ছবি আমরা শূন্য নম্বর দিনে দেখিয়েছি। এই চিপ বা চিপপুঞ্জই ভৌত ড্রাইভটায় তথ্য এবং বিদ্যুৎপ্রবাহের যাতায়াতকে নিয়ন্ত্রণ করে। অপারেটিং সিস্টেমের আদেশ মোতাবেক ড্রাইভকে সক্রিয় করে তোলে এই কন্ট্রোলার এবং আদেশ-নির্দিষ্ট তথ্য ফ্লপি-ডিস্ক এবং সিপিইউ-র মধ্যে দেওয়া নেওয়া করে। যখন ফ্লপি-ডিস্কে তথ্য লেখা হয় তখন তথ্যটা যায় সিপিইউ থেকে ফ্লপি-ড্রাইভ, আর যখন ফ্লপি-ডিস্ক থেকে তথ্য পড়া হয়, তখন তথ্য যায় ফ্লপি-ড্রাইভ থেকে সিপিইউ। নিয়ন্ত্রণ করার জন্যে এই কন্ট্রোলার সার্কিট এবং মূল ভৌত যন্ত্রাংশটা — এই দুটো জিনিষ মিলে তৈরি হয় মেশিনের ওই ড্রাইভটা। শুধু ফ্লপি না প্রতিটি ড্রাইভের বেলাতেই তাই।

ড্রাইভের আদত নিয়ন্ত্রণটা অত্যন্ত জটিল গোলমেলে এবং ঘোরপ্যাঁচসঙ্কুল ইলেকট্রনিক একটা ব্যাপার। তাকে একটা সহজতর আকারে অপারেটিং সিস্টেমের কাছে হাজির করে এই কন্ট্রোলার। যেমন, আপনি হয়ত হার্ডডিস্ক থেকে একটা বিশেষ নামের ফাইল খুলতে চাইলেন, পড়বেন বলে। অপারেটিং সিস্টেম তার আইনোড তালিকা পড়ে সেই ফাইলের নামের সঙ্গে মিলিয়ে নিল, তারপর হার্ডডিস্ক ডিভাইস কন্ট্রোলারকে জানাল, এত এত নম্বর সেক্টর থেকে তথ্য পড়ে আমায় পাঠাও। হার্ডডিস্ক ডিভাইস কন্ট্রোলার তথ্যটা পড়ে সিপিইউ-র কাছে পাঠিয়ে দিল। অপারেটিং সিস্টেম তখন সিপিইউ-কে দিয়ে সেই তথ্যটা ফের পাঠিয়ে দিল কনসোলের ডিভাইস কন্ট্রোলার মানে ভিডিও কন্ট্রোলারের কাছে। ভিডিও কন্ট্রোলারের নিয়ন্ত্রণে ভিডিও ডিভাইস তখন সেই তথ্যটা তার শরীরে মানে কনসোলে ফুটিয়ে তুলল, এবং আপনি ফাইলটা পড়লেন।

এই গোটা কাজটার ভিতরে নিহিত জটিলতাটাকে খেয়াল করুন। ডিভাইস কন্ট্রোলারগুলোর কল্যাণে গোটা কাজটা কত সহজ হয়ে যাচ্ছে। অপারেটিং সিস্টেমের পাঠানো ওই সেক্টর নম্বরটা থেকে কন্ট্রোলার হিশেব কষে ফেলল এত নম্বর সেক্টর মানে কত নম্বর সিলিন্ডার, কোন জায়গা, কোন হেডকে কতটা ঘুরিয়ে সেখানে পাঠাতে হবে — ইত্যাদি (এই সিলিন্ডার ইত্যাদি আর একটু বিশদ ভাবে আমরা জানব যখন ফাইল সিস্টেমের আলোচনায় ঢুকব, মানে আট নম্বর দিনে গিয়ে)।

এখানে জটিলতা আরো একটু বাড়তে পারে এই ভাবে যে হার্ড ডিস্কে হয়ত কিছু ব্যাড সেক্টর দেখা দিয়েছে, যেখানে আর তথ্য পড়া বা লেখা যাবেনা, সেই সেই সেক্টরের তথ্য তুলে নিয়ে হার্ডডিস্ক কন্ট্রোলার অন্য নির্দিষ্ট কিছু সেক্টরে রেখেছে। এই হিশেবটাও মাথায় রাখতে হবে কন্ট্রোলারকে। এবার তাকে হিশেব করতে হবে ওই হেডকে ওই জায়গায় পাঠানো মানে কতটা নড়ানো, কতটা বিদ্যুৎপ্রবাহ লাগবে তাতে — ইত্যাদি। ফাইল পড়ার পুরো প্রক্রিয়াটা জুড়েই নিরবচ্ছিন্ন ভাবে এই কাজটা করে চলতে হবে কন্ট্রোলারকে। এবং একই সঙ্গে, যত যত বিট সে পড়ছে — সেই তথ্যটাকে ওই কম্পিউটারের ওয়ার্ডের মাপে দুই বা চার বা আট বাইট করে পাঠাতে থাকবে অপারেটিং সিস্টেমের কাছে। এই গোটা কাজটা করে চলে কন্ট্রোলারের সার্কিট বোর্ডের যাবতীয় উপাদানগুলো সকলে মিলে। ডিভাইস কন্ট্রোলারগুলো না-থাকলে তাই ওএস-এর কাজটা বহুগুণ জটিলতর হয়ে উঠত।

যা বললাম, এই কন্ট্রোলারকে বাদ দিয়ে ওই পেরিফেরাল ডিভাইসের আর একটা অংশ হল ডিভাইসের নিজেরই ভৌত শরীরটা। কন্ট্রোলারের তুলনায় ডিভাইসের নিজের জটিলতাটা বরং অনেক কম। তার একটা কারণ যে তাদের খুব একটা কিছু নানা ধরনের কাজ করতে হয়না, আর দ্বিতীয় কারণ, এই ধরনের ডিভাইসগুলোর সমস্ত নির্মাতাকেই কিছু স্বীকৃত মান বা স্ট্যান্ডার্ড মেনে নিতে হয়। যে স্ট্যান্ডার্ড অনুযায়ী এদের বানানো হয় — নইলে আলাদা আলাদা কোম্পানির ডিভাইস আলাদা আলাদা মেশিনে কাজ করানো যাবেনা। যেকোনো কোম্পানির আইডিই (IDE — **I**ntegrated-**D**rive-**E**lectronics) ডিভাইস যাতে অন্য যে কোনো কোম্পানির কন্ট্রোলার-এর সঙ্গে লাগানো যায়। পেন্টিয়াম অ্যাথলন এই জাতীয় সমস্ত মেশিনেই লাগানো হয় এই আইডিই ডিস্ক — আমরা যে ধরনের হার্ডডিস্ক নিয়েই সচরাচর কাজ করি। মূল ডিভাইসের ভিতরকার জটিলতাটাকে আমাদের প্রায় কখনোই আলাদা করে মুখোমুখি হতে হয়না এই কন্ট্রোলারের দৌলতে।

এটা হল আইও ডিভাইসের হার্ডওয়ার নিয়ন্ত্রণ, এবার একটু আলগা ছুঁয়ে নেওয়া যাক এর সফটওয়ার নিয়ন্ত্রণের প্রসঙ্গ। হার্ডওয়ারের স্তরে, ইলেকট্রনিক ইলেকট্রিকাল এবং ফিজিকাল, বৈদ্যুতিন বৈদ্যুতিক এবং ভৌত স্তরে, প্রতিটি ডিভাইস এবং তার কন্ট্রোলারের গঠন অন্য প্রতিটি ডিভাইসের থেকে আলাদা। তাই স্বাভাবিকভাবেই এদের কাজ করার জন্যে লাগবে আলাদা প্রোগ্রাম বা সফটওয়ার। একেই বলে ডিভাইস ড্রাইভার। মাউসের জন্যে মাউস ড্রাইভার, কনসোলের জন্যে ভিডিও ড্রাইভার ইত্যাদি। এই সফটওয়ার এবার সরাসরি কথা বলে ওই কন্ট্রোলারের সঙ্গে। কন্ট্রোলার কার্ড যারা বানায় তারাই বানায় এই ড্রাইভার, বা অপারেটিং সিস্টেমের নিজের ইনস্টলার-এর সঙ্গেই দেওয়া থাকে এগুলো। আলাদা আলাদা অপারেটিং সিস্টেমের জন্যে আলাদা আলাদা ড্রাইভার।

আইও ডিভাইসগুলোকে তাদের ভৌত উপাদান, এবং সেই ভৌত উপাদানকে কাজে লাগানোর জন্যে কন্ট্রোলার তথা ডিভাইস ড্রাইভারের সফটওয়ারের কার্যধারার ভিত্তিতে মোটামুটিভাবে দুটো বড় ভাগে ভাগ করা যায়।

ব্লক ডিভাইস — তথ্য-আয়তন-ভিত্তিক উপাদান। এরা তথ্যকে নাড়াচাড়া করে কিছু নির্দিষ্ট আয়তনের এককে, যে এককটার নাম ব্লক। এই প্রতিটি ব্লকের নিজস্ব একটা ঠিকানা আছে, সূচক আছে — মানে অন্য ব্লক তথা তথ্য থেকে স্বাতন্ত্র আছে। সচরাচর এই ব্লকের মাপ হয় ৫১২ থেকে ৩২৭৬৮ বাইট। এর সবচেয়ে গুরুত্বপূর্ণ বৈশিষ্ট্য এই যে, কোনো একটা নির্দিষ্ট ব্লক থেকে স্বতন্ত্র ভাবে তথ্য পড়া যায়, বা সেখানে লেখা যায়, অন্য ব্লকগুলোর লেজে কোনো পাড়া না-দিয়েই। অপারেটিং সিস্টেম চাইলে এক তিন পাঁচ এবং চৌষট্টি নম্বর ব্লকে তথ্য লিখতে পারে দুই চার এবং ছয় থেকে তেষট্টিকে নিষ্কলুষ শূন্য রেখে দিয়ে। ব্লক ডিভাইস নিয়ে এবং ব্লক এককে তার তথ্য নাড়াচাড়ার বিষয়টা নিয়ে আমরা অনেকটা আলোচনা করব আট নম্বর দিনে গিয়ে, ফাইলসিস্টেম নিয়ে আলোচনায়।

ক্যারেকটার ডিভাইস — তথ্য-চিহ্ন-ভিত্তিক উপাদান। ক্যারেকটার ডিভাইসের কাছে তথ্য মানে সারিবদ্ধ কিছু চিহ্ন বা ক্যারেকটারের প্রবাহ, এদের মোট আয়তন বা ব্লক সাইজ কী তা নিয়ে ক্যারেকটার ডিভাইসের বিন্দুমাত্র এসে যায়না। এখানে তথ্যের কোনো স্বতন্ত্র ঠিকানা বা অস্তিত্ব নেই। তাই এখানে আলাদা করে কোনো তথ্য একককে খোঁজা বা চিহ্নিত করা যায়না। প্রিন্টার, মোডেম, ল্যান কার্ড, মাউস জাতীয় পয়েন্টিং ডিভাইস — এরা সবাই হল ক্যারেকটার ডিভাইস। ডিস্ক জাতীয় নয় যে আইও ডিভাইসগুলো তারা সকলেই ক্যারেকটার ডিভাইস।

একটু ভালো করে ভাবলেই বোঝা যায় এই দুটো ধরনের আইও ডিভাইসের মধ্যে বর্গীকরণটা কোনো জল-অচল বর্গ বা ওয়াটারটাইট কম্পার্টমেন্ট নয়। ধরুন হার্ডডিস্ক যে ব্লক পদ্ধতিতে কাজ করে এটা খুব সহজেই বোঝা যায়। একটা হার্ডডিস্কের ভিতরে একটা হেড একটা সিলিন্ডারের গায়ে যে সেক্টরেই থাকুক না কেন সেই হেডটাকে নড়িয়ে বা অন্য কোনো হেডকে কাজে লাগিয়ে সহজেই অন্য একটা ব্লকে তথ্য লেখা যায় বা সেখান থেকে পড়া যায়। আলাদা আলাদা ঠিকানার ব্লককে চিনে নেওয়া যায় সহজেই।

এই সহজতাটা ঘেঁটে যায় যদি আপনি ক্যাসেট ড্রাইভকে ভাবেন। যেখানে একটা ক্যাসেটের ফিতেয় তথ্য তুলে রাখা হচ্ছে, বা সেখান থেকে পড়া হচ্ছে। সতত এই ক্যাসেট ড্রাইভ একটা ক্যারেকটার ডিভাইস হিশেবেই বিবেচ্য — প্রবহমান চিহ্নরাশির বীচিবিভঙ্গ খচিত করে রেখে চলেছে তার চৌম্বক শরীরে। এবার ভাবুন তো, একটা গোটা টেপময় তথ্যের ভাঁড়ারকে তো একটা একরৈখিক ব্লকের মিছিল বলে ভাবা যেতেই পারে — অপারেটিং সিস্টেম যখনই একটা বিশেষ অমুকতম ব্লকে তথ্য পড়তে বা লিখতে চাইবে, আরামসে সেটা সে করতে পারে, ক্যাসেটের ফিতেটাকে এগিয়ে পিছিয়ে, রিওয়াইন্ড আর ফাস্ট-ফরওয়ার্ড করে, যতক্ষণ না কোনো অনির্দেশ্য তমুক নয়, নিশ্চিতভাবেই ওই বিশেষ অমুকতম ব্লকটিকে সে খুঁজে পাচ্ছে। তার মানে ঠিক হার্ডডিস্কের মতই স্বতন্ত্র ঠিকানার ব্যক্তিস্বাতন্ত্র নিয়ে মুক্ত ব্লকতন্ত্র মানে ব্লকের গনতন্ত্র গড়ে তোলা যাচ্ছে ক্যাসেটের ফিতেতেই, শুধু একটু ইয়ে, মানে সময়টা একটু বড্ডই বেশি লাগছে। তাহলে ক্যারেকটারের ক্যারেকটার রইল কই — সে তো ব্লক হয়ে গেল?

কিন্তু এ প্রায় নব্যন্যায় হয়ে যাচ্ছে — বাঙালীর বিখ্যাত সেই তীব্র মেধাদীপ্ত সিরিয়াস ভাটের ট্র্যাডিশনের মত, বাঙালীর এই ক্যালি বোধহয় সম্রাট অশোকও জানতেন, ব্রাহ্মী শিলালিপিতে আছে গৌড়ীয় রীতির মানে কথাকে ফেনিয়ে বলার প্রথার উল্লেখ, বাজে বকাটা বোধহয় আমাদের বংশগত জাতিগত ইতিহাসগত — একটা যুক্তিকে তার লিমিটে পুশ করা — ক্যাসেট ড্রাইভ তো আর সত্যিসত্যিই ওভাবে ব্যবহার করা হয়না।

এটা ঠিক যে এই বর্গীকরণটা নিখুঁত নয়। কোনো কোনো ডিভাইস যেমন, ব্লক বা ক্যারেকটার কোনো খাপেই ঢোকানো যায়না। ক্লক বা ঘড়ি যেমন। এদের ব্লক-ঠিকানা দিয়ে ধরা যায়না, আবার এরা কোনো চিহ্নপ্রবাহ-ও পয়দা করেনা, ঘড়ি কি তবে কেউ নয়? সে তো যা করে সেটা হল কিছুক্ষণ অন্তর অন্তর একটা করে ইন্টেরাপ্ট মানে অপারেটিং সিস্টেমের উদ্দেশ্যে একটা করে সিগনাল তৈরি করে যাওয়া। তাহলে? এই ব্লক বা ক্যারেকটার ডিভাইস হিশেবে বর্গীকরণটা এইজন্যেই আছে যে এই ছকটা দিয়ে অপারেটিং সিস্টেমের আইও ডিভাইসদের নাড়াচাড়া করার প্রকৌশলগুলো বেশ ভালো ব্যাখ্যা করা যায়। এই কথাগুলো আনতে হল কারণ, পরে যখন আমরা গ্নু-লিনাক্স অপারেটিং সিস্টেমের ডিভাইস ফাইল এবং ফাইলসিস্টেম প্রসঙ্গে আলোচনা করব তখন এই পার্থক্যটা আমাদের দরকার পড়বে। ফাইলসিস্টেম যেমন ব্লক ডিভাইসগুলোকে নিজের আভ্যন্তরীন করে, কিন্তু ক্যারেকটার

ডিভাইসগুলোকে ছেড়ে দেয় আরো নিচের স্তরের সফটওয়্যারদের হাতে ছেড়ে দেয় — আমরা পরে আসব এই নিয়ে দীর্ঘ আলোচনায়।

ডিভাইস	তথ্যহার/সেকেন্ড
কিবোর্ড	১০ বাইট
মাউস	১০০ বাইট
৫৬-কে মোডেম	৭ কেবি
টেলিফোন লাইন	৮ কেবি
লেজার প্রিন্টার	১০০ কেবি
স্ক্যানার	৪০০ কেবি
ইথারনেট	১.২৫ এমবি
আইডিই ডিস্ক	৫ এমবি
৪০ x সিডিরম	৬ এমবি

আইও নিয়ে আমাদের এই আলোচনাটা আপাতত শেষ হল, বিভিন্ন ডিভাইসের তথ্য দেওয়া-নেওয়া হারের একটা তালিকা দিয়ে। অপারেটিং সিস্টেমকে তৈরি থাকতে হয় এই আলাদা আলাদা গতিবেগগুলোর প্রত্যেকটার সঙ্গে মানিয়ে নেওয়ার। বিভিন্ন আইও ডিভাইসের এবং বাস-সংযোগের এই বিট-রেট বা তথ্য বহন করার হার — এটা কিন্তু রোজ বদলাচ্ছে। এটা আমি তুলছি জুলাই ২০০২-এ ছাপা অ্যান্ড্রু ট্যানেনবমের অপারেটিং সিস্টেম বইটা থেকে। সায়মিন্দু কাল প্রস্তাব দিল, যে এই পাঠমালাটা অঙ্কুর বাংলা-লাইভ সিডিতে রাখবে — তার মানে আপনার পড়তে পড়তে আরামসে ২০০৪ হয়ে যাচ্ছে — এর মধ্যে কোথাকার বিট কোথায় গড়াবে কে জানে।

৭।। আইও তথ্য, ইন্টেরাপ্ট, ডিএমএ

প্রতিটি কন্ট্রোলারের মধ্যে থাকে কয়েকটা করে রেজিস্টার। রেজিস্টার তো আমরা চিনি, সিপিইউ-র মধ্যে থাকে যেমন, এক নম্বর দিনে আলোচনা হয়েছে। অপারেটিং সিস্টেম যখন কোনো আইও ডিভাইস নিয়ে কিছু করতে চায়, তাকে কথা বলতে হয় এই ডিভাইস কন্ট্রোলারের সঙ্গে আর সেই কথোপকথনের দোভাষীর কাজ করে এই রেজিস্টারগুলো। একটা হার্ডডিস্ক কন্ট্রোলারে যেমন সচরাচর রেজিস্টার থাকে ডিস্কের শরীরে ব্লক-ঠিকানা চিহ্নিত করার, মেমরির শরীরে বাইট-ঠিকানা চিহ্নিত করার (এক নম্বর দিনের কূট কিন্তু আপাতঅগ্রাহ্য সেকশন থেকে একটু দেখে নিন মেমরি ঠিকানার কনসেপ্টটা, যদি ভুলে গিয়ে থাকেন), সেক্টর গোনার, এবং তথ্যের গতিমুখের — মানে ঢুকছে না বেরোচ্ছে, তথ্য লেখা হচ্ছে না পড়া হচ্ছে, ইত্যাদি। আমরা আগেই বলেছি, ডিভাইসকে হার্ডওয়ার নিয়ন্ত্রণ করে কন্ট্রোলার, আর এই কন্ট্রোলারকে তথা ডিভাইসকে সফটওয়্যার নিয়ন্ত্রণ করার জন্যে থাকে একটা ডিভাইস ড্রাইভার যা কারনেল বা অপারেটিং সিস্টেমের সঙ্গে যুক্ত হয়ে যায়, যাতে অপারেটিং সিস্টেম ওই ডিভাইসের রসদটাকে বন্টন করতে পারে অ্যাপ্লিকেশন প্রোগ্রামগুলোর ভিতরে। এবার, যখন অপারেটিং সিস্টেম একটা ডিভাইসকে কাজে লাগাতে চায়, একটা সিগনাল পাঠায় এই ডিভাইস ড্রাইভারের কাছে। এই সিগনালকে তখন কন্ট্রোলারের কাছে পাঠিয়ে দেয় ডিভাইস ড্রাইভার ওই রেজিস্টারগুলোর মারফত।

একদিকে সিপিইউ, আর অন্যদিকে একদম ভৌত স্তরের ডিভাইস — এই দুই মেরুর মধ্যে চলাচল করে তথ্য, আইও বা ইনপুট আউটপুট তথ্য। সেটাই কাজ আইও ডিভাইসগুলোর, বাইরের তথ্য সিপিইউতে আনা, এবং সিপিইউ-র তথ্য বাইরে নিয়ে যাওয়া। এই তথ্য যাতায়াতের কাজটা মূলত ঘটে তিন প্রক্রিয়ায়। তিনটে পদ্ধতির নাম হল ব্যস্ত-অপেক্ষা বা বিজি-ওয়েটিং, ইন্টেরাপ্ট বা ব্যাঘাত, এবং ডিএমএ (DMA — **D**irect-**M**emory-**A**ccess) বা সরাসরি-স্মৃতি-সংযোগ। একটু আলতো করে ছুঁয়ে আসা যাক এই তিনটে আইও তথ্য যাতায়াতের রকমকে। এর মধ্যে ইন্টেরাপ্ট আর ডিএমএ-র প্রসঙ্গ গোটা পাঠমালা জুড়েই বারংবার আসবে।

৭.১।। ব্যস্ত-অপেক্ষা বা বিজি-ওয়েটিং

আইও তথ্য যাতায়াতের সবচেয়ে সরল পদ্ধতি হল বিজি-ওয়েটিং। ইউজারের চালানো একটা প্রয়োগ বা অ্যাপ্লিকেশন সফটওয়ার, তার কাজের সূত্রে, কিছু পরিমাণ তথ্য নড়াতে চাইল এক জায়গা থেকে আর এক জায়গা। ডিভাইস থেকে ফাইল পড়া হতে পারে, ডিভাইসে ফাইল লেখা হতে পারে, ডিভাইস-মেমরি-সিপিইউ এদের মধ্যে কাজগত নাড়াচাড়াও হতে পারে। প্রয়োজন পড়া মানে, অ্যাপ্লিকেশন প্রোগ্রাম এবার ওএস মানে কারনেল মানে চূড়ান্ত রসদ-মঞ্জুরি কমিশনের কাছে আর্জি জানাল। নিজে নিজে রসদ উপাদান ব্যবহার করে নেওয়ার তো তার অধিকার নেই, আগেই বলেছি। এবার, অ্যাপ্লিকেশন প্রোগ্রাম ওএস বা কারনেলের কাছে তার রসদ ব্যবহারের আর্জি জানানো মাত্র, এই আর্জিটার চালু টেকনিকাল নাম হল সিস্টেম-কল (system-call), সেই সিস্টেম-কলটাকে কারনেল অনুবাদ করে নিল একটা আদেশে, যা ডিভাইস ড্রাইভারের কাছে কারনেল পাঠাবে। ডিভাইস ড্রাইভারের কাছে কারনেলের পাঠানো এই আদেশটার টেকনিকাল নাম হল প্রসিডিওর কল (procedure-call)।

ওএস-এর কাছ থেকে প্রসিডিওর কল আসা মাত্র ড্রাইভার এবার চালু করে তোলে ওই ডিভাইসকে তথা ডিভাইসের মাধ্যমে আইও তথ্য পড়া বা লেখার কাজ, এবং নির্নিমেষ পর্যবেক্ষণ করে চলে সক্রিয় ডিভাইসটাকে। আর বাধ্য ডিভাইস তার ড্রাইভারের আদেশানুযায়ী আইও তথ্যের কাজ করে চলে। যেই আইও তথ্য পড়া/লেখার কাজটা শেষ হয়, ড্রাইভার তখন প্রসিডিওর কল মোতাবেক তার যেখানে যেখানে তথ্য পাঠানোর আছে — সেই গোটা কাজটা শেষ করে ফেরত আসে তার স্বাভাবিক নিষ্ক্রিয় অবস্থায়। ওএস বা কারনেলও ফেরত যায় অ্যাপ্লিকেশন প্রোগ্রাম থেকে সিস্টেম কল আসার আগের পুরোনো অবস্থায়। এই সিস্টেম-কল থেকে প্রসিডিওর-কল থেকে ডিভাইস ড্রাইভার তথা ডিভাইসের সক্রিয়তা থেকে পর্যবেক্ষণ থেকে ফের পুরোনো নিষ্ক্রিয়তা — এই গোটা লুপটা শেষ হয়। এটাই ব্যস্ত-অপেক্ষা বা বিজি-ওয়েটিং পদ্ধতি।

পর্যবেক্ষণটা ঘটে একটা লুপ বা পুনরাবৃত্ত পথে — এক নম্বর দিনের ওই fetch … repeat লুপটা মনে করুন, অনেকটা ওই রকম। দেখে, বসে থাকে, দেখে . . . এইরকম চলতেই থাকে। যতক্ষণ না ওই ইনপুট/আউটপুটের কাজ শেষ হয়। যেই শেষ হয় তখনি ড্রাইভার তথ্যটাকে যেখানে দরকার সেখানে পাঠিয়ে দেয় এবং ফের নিষ্ক্রিয় হয়ে যায়। অপারেটিং সিস্টেম আবার ইউজার প্রোগ্রামটাকে জানিয়ে দেয়, তোমার কাজ শেষ হয়ে গেছে। এই কায়দার অসুবিধেটা হল এই যে কাজের গোটা সময়টা জুড়ে ওই লুপে আটকে থাকতে হয় ডিভাইস কন্ট্রোলারকে, আর তাই আটকে থাকতে হয় ডিভাইস ড্রাইভার দিয়ে কন্ট্রোলারকে যে চালাচ্ছে সেই সিপিইউ বা প্রসেসরকেও। সেই অর্থে এই পদ্ধতিটা তাই ইনএফিশিয়েন্ট বা ক্যাবলা, একটা কম্পিউটারের সবচেয়ে দামী এবং জরুরি জিনিষটাকে আটকে রাখে তার খ্যাঁচাকলে।

৭.২।। ইন্টারাপ্ট বা ব্যাঘাত পদ্ধতি

দু-নম্বর কায়দাটায় অ্যাপ্লিকেশন প্রোগ্রামের সিস্টেম-কল কারনেলের মাধ্যমে প্রসিডিওর কলে পরিণত হয়ে ডিভাইস ড্রাইভারের কাছে যায়। ডিভাইস ড্রাইভার ডিভাইসকে চালু করে দেয়, করে জানিয়ে দেয়, কাজ শেষ হলে ভাই আমাকে একটা ফোন করে দিস। এই ফোনটাই হল ইন্টেরাপ্ট। ডিভাইসটার ভৌত শরীরে ঘটমানতার বিষয়ে ডিভাইস ড্রাইভারের কারনেলের কাছে পাঠানো সিগনাল। এই সিগনালটা পাঠিয়ে দিয়ে ডিভাইস কন্ট্রোলার ফেরত চলে আসে তার পুরোনো অবস্থায়, কারনেলের কাছ থেকে ডিভাইস ড্রাইভার মারফত নতুনতর আদেশ আসা অব্দি ঘুমোতে থাকে। অপারেটিং সিস্টেম এবার যে ইউজার প্রোগ্রাম রসদ ব্যবহারের ওই আর্জিটা জানিয়েছিল তার কাছ থেকে নতুনতর কোনো আর্জি আসার পথটা আপাতত বন্ধ করে দিয়ে চোস্ত পাজামা আর গোড়ালি অব্দি লম্বা পাঞ্জাবি পরে রাজনৈতিক রোঁদে বেরোয় — আর কোনো প্রোগ্রামের কোনো আর্জি আছে কিনা, দেখতে হবেনা জনগণ কেমন আছে। আর কারুর জন্যে কিছু করতে হবে ভাই? ডিভাইস কন্ট্রোলার ওদিকে তার ডিভাইসের কাজ শেষ হলে যথারীতি কথামতো একটা ফোন করে দেয় ড্রাইভারের কাছে — ইন্টেরাপ্ট। কারনেলের কাছে খবর যায়, ওদিকটা ফ্রি হয়ে গেছে দাদা। আবার কোনো আর্জি নিতে পারো। এই প্রোগ্রামের হোক, বা অন্য প্রোগ্রামের।

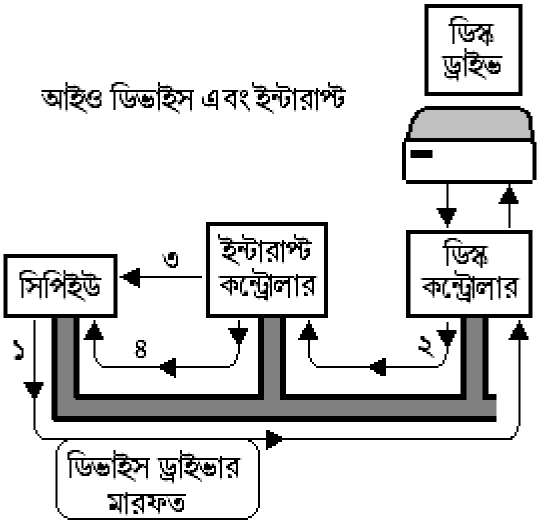
ওএস বা অপারেটিং সিস্টেমের কার্যপ্রনালীতে ইন্টারাপ্ট বা ব্যাঘাত একটা জরুরি ভূমিকা পালন করে। ছবিতে দেখুন, আমরা তিনটে আবশ্যিক এবং একটা সম্ভাব্য ধাপে ভেঙে গোটা আইও তথ্য চলাচল এবং ইন্টারাপ্টের পুরো পদ্ধতিটা দেখিয়েছি।

ধাপ ১।

সিপিইউ ডিভাইস ড্রাইভার মারফত তার আদেশ পাঠাচ্ছে ডিস্ক কন্ট্রোলারকে। মানে ডিভাইস কন্ট্রোলারের রেজিস্টারগুলোতে দেগে দিচ্ছে — তার ঠিক কী চাই। কন্ট্রোলার এবার চালু করে দিচ্ছে ভৌত ডিভাইসটাকে। সে একদম ভৌত স্তরে তার তথ্য লেখার বা পড়ার কাজে ব্যস্ত হয়ে যাচ্ছে।

ধাপ ২।

কন্ট্রোলার গোটা সময়টা জুড়ে তার ঘনিষ্ঠ পাহারাটা চালিয়ে যাচ্ছে ডিভাইসের উপর। যেই, সিপিইউর আদেশমত বাইট স্থানান্তরের গোটা কাজটা সাঙ্গ হয়ে যাচ্ছে, কন্ট্রোলার সেই সংবাদটা জানিয়ে দিচ্ছে ইন্টেরাপ্ট কন্ট্রোলার চিপকে। ঠিক ডিভাইস গুলোর যেমন কন্ট্রোলার চিপ এবং সার্কিট থাকে, ইন্টারাপ্ট বা ব্যাঘাতের গোটাটা নিয়ন্ত্রণ করার জন্যেও একটা নিয়ন্ত্রক সার্কিট থাকে। ডিভাইস কন্ট্রোলার এই সংবাদটা ইন্টারাপ্ট কন্ট্রোলারকে পাঠাচ্ছে তথ্য চলাচলের জন্যে কম্পিউটার জুড়ে ছড়ানো যোগাযোগপথ বাসের ভিতরেই বিশেষ কিছু লাইন বেয়ে।



ধাপ ৩।

ইন্টেরাপ্ট কন্ট্রোলার কিন্তু সেই মুহূর্তে ইন্টেরাপ্ট নেওয়ার অবস্থায় থাকতেও পারে, আবার নাও পারে, যদি তার হাতে আরো জরুরি কোনো ইন্টেরাপ্ট থাকে। এই মুহূর্তে যদি তার ইন্টেরাপ্ট নেওয়ার অবস্থা থাকে, সে ইন্টেরাপ্টটা গ্রহণ করে এবং ইন্টেরাপ্টের ভিতরে থাকা আইও তথ্য স্থানান্তরের কাজ শেষ হওয়ার সিগনালটা পাঠিয়ে দেয় সিপিইউর কাছে।

ধাপ ৪।

এই ধাপটা তখনই ঘটে, যদি একাধিক আইও ডিভাইস একই সঙ্গে চালু অবস্থায় থাকে। তখন ইন্টেরাপ্ট কন্ট্রোলার যে ডিভাইসের তথ্য স্থানান্তরের কাজ হল, সেই ডিভাইসের নম্বরটা বাসে তুলে দেয়, যাতে সেখান থেকে পড়ে কারনেল কোন আইও ডিভাইস ফাঁকা হয়েছে। এই নম্বর দিয়েই কারনেল একটা ডিভাইসকে চেনে। ছয় নম্বর দিনে আমরা এই ডিভাইস এবং তার নম্বরের বিষয়টা জানব, ডিভাইস ফাইলের সূত্রে। আমরা দেখব, গ্নু-লিনাক্সে সবকিছুই ফাইল, এমনকী ডিভাইসগুলোও।

ডিএমএ বা সরাসরি-স্মৃতি-সংযোগ পদ্ধতি

আইওর তৃতীয় কায়দাটা কাজ করে একটা ডিএমএ চিপ (DMA — **D**irect-**M**emory-**A**ccess) মারফত। এই ডিএমএ চিপটা সরাসরি ডিভাইস কন্ট্রোলার আর মেমরির মধ্যে তথ্যের যাতায়াত নিয়ন্ত্রণ করে। আইও তথ্যের স্থানান্তরের উপর অহর্নিশ বাধ্যতামূলক সিপিইউ-র খবরদারিটা আর ঘটেনা এই পদ্ধতিতে। এখানে সিপিইউ ডিএমএ চিপকে জানিয়ে দেয় — কতগুলো বাইট নড়াতে হবে, ডিভাইসের ঠিকানা কী, মেমরির ঠিকানা কী, এবং কোনদিকে যাবে তথ্যটা, ডিভাইস থেকে মেমরি না মেমরি থেকে ডিভাইস। ডিএমএ চিপের কাজ যেই শেষ হয়, সে অমনি একটা ইন্টেরাপ্ট পাঠায়। অপারেটিং সিস্টেম টের পায় কাজ শেষ হয়ে গেছে।

অপারেটিং সিস্টেমের কাছে পাঠানো এই ইন্টেরাপ্টগুলোকে নাড়াচাড়া করার জন্যে যে ইন্টেরাপ্ট কন্ট্রোলারটা থাকে, সে কোনো একটা মুহূর্তে কোনো একটা ডিভাইস থেকে আসা ইন্টেরাপ্টকে চাইলে স্থগিত করে দিতে পারে, বা নতুন করে সক্রিয় করে তুলতে পারে, কারণ, প্রায়ই ইন্টেরাপ্ট কন্ট্রোলারের কাছে একই সঙ্গে একাধিক ইন্টেরাপ্ট এসে পড়ে। তখন কোন ইন্টেরাপ্টকে আগে মনোযোগ দেবে ইন্টেরাপ্ট কন্ট্রোলার সেটা ঠিক হয় প্রায়োরিটি বা প্রাথমিকতা দিয়ে। কোন ইন্টেরাপ্টটা বেশি জরুরি। পরে, যখন প্রসেস এবং প্রায়োরিটি নিয়ে, পদ্ধতি এবং তার প্রাথমিকতা নিয়ে আমরা আলোচনা করব তখন এই ধারণাটা আমাদের দরকার পড়বে। আর এই সেকশনেই আমরা যে কথাগুলো বলেছিলাম মাল্টিপ্লেক্সিং নিয়ে, পরে আরো বলব, তার সঙ্গে মিলিয়ে নিন একে। এই কালগত মাল্টিপ্লেক্সিং, ভূমিগত মাল্টিপ্লেক্সিং, এই লাইনেই আসবে টাইমশেয়ারিং, মাল্টিপ্রোগ্রামিং, মাল্টিটাস্কিং-এর আলোচনা। চার এবং পাঁচ নম্বর দিনে।

৮।। বাস-যোগাযোগ

আজকের আলোচনারই একদম গোড়ায় আমরা একটা বাস-যোগাযোগব্যবস্থার ছবি দেখিয়েছি। এটা হল একটা সরল বাস। এখনকার বাস এর চেয়ে জটিল হয়। এই সহজতর বাস ব্যবহৃত হত একদম গোড়ার দিককার আইবিএম পিসিতে। পাঁচ নম্বর দিনে এ নিয়ে আরো বিশদ আলোচনা আসবে, তবু মূল কথাটা এই যে ক্রমে প্রসেসরগুলোর গতিবেগ বাড়ছিল, মেমরিদেরও। এই নতুন ধরনের প্রসেসর আর মেমরি কাজে লাগিয়ে তথ্যের ট্রাফিকের মোট পরিমাণটাও বাড়ছিল হুহু করে। তার সঙ্গে আর পাল্লা দিতে পারছিলনা এই সরল এক রাস্তার বাসব্যবস্থা। তাই এর সঙ্গে আরো আরো বাস যোগ করা হচ্ছিল। একদিকে সিপিইউ আর মেমরির মধ্যে। অন্যদিকে নতুন যুগের দ্রুততর আইও ডিভাইসগুলোর সঙ্গে তাদের কন্ট্রোলারদের তথা সিপিইউর। তবে তাদের আলোচনা করতে গেলে যে পরিমাণ হার্ডওয়ার আলোচনা আনতে হবে সেটা খুব একটা পছন্দ হচ্ছেনা। আর সেগুলো বোঝানো মানে নতুন ছবি আঁকা, আর ডাক্তারের আদেশ মোতাবেক আমাকে বসতে হচ্ছে ডান পাটা উপরে তুলে, টেবিলের পাশে খাটে একটা জলচৌকি রেখে। এই অবস্থায় মাউস চালাতে অসুবিধে হচ্ছে খুব। কোমরে লাগছে। ডাক্তার বলেছে ভেঙেছে পায়ের পাতার মেটাটার্সাল বা এই গোছের দুটো হাড়। অথচ ব্যাথা করছে গোড়ালি হাঁটু হয়ে কোমর অব্দি। আমার পায়ের পাতাটা যে আমার কোমর অব্দি এটা আমার আগে জানা ছিলনা। এত ব্যথাট্যথা নিয়ে বাসজার্নি ভালো না। ব্যস, বাস এখানেই শেষ। এর পরে সিস্টেম লিখতে গিয়ে খুব প্রয়োজন পড়লে একটু ছোট করে মেরে দেওয়া যাবে। পা ভাঙল বুধবার, বারোই নভেম্বর রাতে। আজ রবিবার। সোয়া চার দিন গেছে, এর মধ্যে এক আর দুই নেমে গেল। অবশ্য প্রথম খসড়া তো ছিলই। জয় পা।

আমাদের এই গ্নু-লিনাক্স ইশকুলের পাঠটাও কাজ করে চলেছে একের পর এক ফাইল জুড়ে, জিএলটি-এলইএস০০, জিএলটি-এলইএস০১, জিএলটি-এলইএস০২, ... কিন্তু সেটা তো শুধু মৃত স্থির পরিবর্তনহীন ফাইল — গতিশীল ফাইল — প্রোগ্রাম থেকে প্রসেস, তাই ফাইল-আবদ্ধ তার আকারটাও এই জ্যান্ত গতির প্রকোপে বদলে যাচ্ছে — গ্নু-লিনাক্সে ভাবতে শেখার সাহায্যের জন্যে লেখা জিএলটি ইশকুল পাঠমালাটা নিজেও জ্যান্ত। আমরা তার জ্যান্ত অংশীদার। তাই ফাইলে ফাইলে দেগে দেওয়া আমাদের স্থির পরিকল্পনাটাও অস্থির হচ্ছে, নিয়ত বদলাচ্ছে। এখনো আমরা অপারেটিং সিস্টেমের ইতিহাসে ঢুকলাম না।

এখনো আমাদের মূল রেফারেন্স ট্যানেনবম-এর মডার্ন অপারেটিং সিস্টেম আর জিএলটির থেকে দেওয়া গ্নু-লিনাক্স রিসোর্স সিডির লিনাক্স ডিকশনারি এবং রেডহ্যাট লিনাক্সের ডকুমেন্টেশনে গ্লসারিটা। এছাড়া আর একটু ভালো করে বাস ইত্যাদি গুলো বুঝতে চাইলে থম্পসন অ্যান্ড থম্পসন-এর হার্ডওয়ার ইন এ নাটশেল তো আছেই। ভালো কথা, আগের দিন যে সিপ্লাসপ্লাস-এর বইয়ের নামটা মনে পড়েনি সেটার নামটা পরে নেটে পেয়েছি। নিল গ্রে-র এ বিগিনার্স সিপ্লাসপ্লাস। অত্যন্ত উমদা বই। ওরিলির। অপারেটিং সিস্টেম আর প্রোগ্রামিং তথা প্রোগ্রামিং ল্যাংগোয়েজ-এর পারস্পরিক যে সম্পর্কটা — সিস্টেম কল, প্রসিডিওর কল — এগুলো খুব ভালো বোঝানো আছে। আর একটা বইতেও খুব ভালো করে আছে, স্টিফেন প্রাটার ইউনিক্স অ্যাডভান্সড প্রোগ্রামিং। পরের বইটার প্রসঙ্গ পরে খুব বেশি করে আসবে, এই পাঠমালার শেষ মানে দশ নম্বর দিনে — যখন আমরা ব্যাশ তথা শেল স্ক্রিপ্ট নিয়ে কথা বলব।

ও, একটা জিনিষ, আমাদের লাগের ইন্দ্র আর সঙ্কর্ষণ — ওদের দুজনেরই ওই কম্পিউটারকে গাধা বলে ডাকার জন্যে লেখা প্রোগ্রামটা পছন্দ হয়নি খুব একটা। একটু বোকাবোকা লেগেছে বোধহয়। ইন্দ্রর সঙ্গে এখনো সামনাসামনি কথা হয়নি, সঙ্কর্ষণ নিজেই বলেছে। আপনাদের যদি এই নিয়ে কিছু বলার থাকে, বা অন্য কিছু নিয়েও, জানান। মতামতের জন্যে অপেক্ষা করে আছি, আরো সেই মতামত যদি আমারে ফেভারে যায়। জিএলটির মেল আইডি তো দেওয়াই আছে, প্রতিটি দিনের শেষে জিএলটির লোগোয়। আর সায়মিন্দু দিন এক-এর একটা ভুল ধরিয়ে দিয়েছে — ল্যান কার্ড কথাটা বলে চলে গেছি, কিন্তু এই অ্যাক্রোনিমটার পিছনে গোটা নামটা দিইনি। লোকাল এরিয়া নেটওয়ার্ক। এই কার্ড দিয়ে কম্পিউটার অন্য একটা ল্যানিত কম্পিউটারের সঙ্গে কথা বলে।

glt-mad@ilug-cal.org

MADHYAMGRAM

সংকলন ও রচনা : মধ্যমগ্রাম জিএলটি-র (glt-mad@ilug-cal.org) তরফে ত্রিদিব সেনগুপ্ত

জিএলটি মধ্যমগ্রাম
glt-mad@ilug-cal.org

শূন্য নম্বর দিনে আমরা একটা পিসির বাইরে থেকে ভিতর দিকে এগোলাম, এক আর দুইয়ে তার গঠন আর কাজ করার এককগুলো নিয়ে একান্ত প্রাথমিক কিছু কথা বললাম — যার প্রায় গোটাটাই বর্তমান, এই বর্তমান, কেবলই বত্রিশ বিট করিছে নির্মাণ, অথচ এখন তো আমরা প্রায় আমাদের পা ৬৪ বিটের দিকে এগোতে শুরু করেছে, পশ্চিমের বাজারে মেইনফ্রেম আর পিসি দুটো অংশেই এসে গেছে। নিশ্চয়ই এখনো বিকট কিছু একটা দাম — আমি জানার চেষ্টাও করিনি। কিন্তু যে আকারে আমরা পিসিকে পাচ্ছি, আমাদের সামনে, এই আকারটা এই আকারটাই যে হয়ে উঠল, তার জ্যান্ত কিছু ক্রমানুবর্তন আছে। আজকের 'দুলকি চাল' শব্দবন্ধে যেমন অনেক আগের কোনো এক পালকি থেকে ঘোমটার বৌটির বা ঘোড়সওয়ারের চালের লুপ্ত মেটাফর, লুপ্ত মেটাফরকে লুপ্ত বলতে হচ্ছে মানেই তো সেটা লুপ্ত নয়, তার লোপপ্রক্রিয়াটাই আমার আজকের বর্তমান, তেমনি, কম্পিউটারের আজকের অবয়বের মধ্যে, সার্কিট থেকে সার্কিটে, ট্রানজিস্টর থেকে ট্রানজিস্টরে তথ্য এককের নড়ে বেড়ানোর মধ্যে সেই ক্রমানুবর্তন সমানে চলছে। মাতৃগর্ভের ওজনহীন ঝুঁঝকো কোমলতার আশ্রয়ের দুশোআশি দিনে আপনি যেমন পেরিয়ে এসেছেন গোটা হোমো সেপিয়েন্স সেপিয়েন্সের অভিযোজন পথ, অমেরুদণ্ডী, সরীসৃপ, উভচর, ঠিক সেই রকম আপনার পিসির এই গোটা কয়েক আয়তাকৃতি এবং একটা আধাবর্তুল ইলেকট্রনিক্সের মধ্যে, তাদের সঙ্গে আপনার সম্পর্কের মধ্যে রয়ে গেছে এই গোটা ক্রমানুবর্তন। এই ক্রমানুবর্তনটাকে আমরা এবার ধরার চেষ্টা করব। এতে একটা বাড়তি আরামও হবে আমাদের, গত তিনটে দিনে কথা বলতে গিয়ে বারবার গ্নু-লিনাক্স আর উইনডোজ এই দুটোকেই অনেকবার পাশাপাশি আনতে হয়েছে, দুটো থেকেই সমান্তরাল উদাহরণ, অনেকটা প্রতিতুলনার মত, এর মধ্যে একটা তুমুল ঘাপলা আছে। মোট গঠন এবং সেই গঠনকে ভাবার প্রক্রিয়ার নিরিখেই, মানে অন্টলজি আর এপিস্টেমলজি দুটো মাত্রা থেকেই এই দুটো কাঠামো এত এত বেশি পৃথক। আমাদের অলৌকিক অঙ্কস্যার হরিনারায়ণবাবু, ক্লাস সেভেনে তিলকূটের গল্প দিয়ে যিনি আসলে লিমিট আর কনটিনিউয়িটি বুঝিয়ে দিয়েছিলেন, তখন বুঝতেও পারিনি, পেরেছিলাম অনেকগুলো বছর পরে, তিনি এই তুলনার অসম্ভাব্যতা বলেছিলেন, চার ইজ টু পাঁচ তো করবি, চার হাতি ইজ টু পাঁচ কাপ চা কর তো — মানে, তুলনা করা যায় শুধু বিমূর্ত সংখ্যার মধ্যে, কিন্তু, খুদা গাওয়া, গ্নু-লিনাক্স বা উইনডোজ দুজনেই তো বেজায় মূর্ত। আর এই মূর্তিমান আলাদাপনাকে কিছুতেই বোঝা যায়না এই ক্রমানুবর্তনটাকে না জেনে। আমরা এক আর দুই নম্বর দিনে বারবার বলেছি অপারেটিং সিস্টেমের কথা — কিন্তু এই অপারেটিং সিস্টেম চিরকালই ঠিক এই রকম ছিল না, মানে, একসময় তো অপারেটিং সিস্টেমই ছিল না। এখনকার চালু অপারেটিং সিস্টেমের বৈশিষ্ট্যগুলোর ঠিক এই এই বৈশিষ্ট্য হয়ে ওঠার মধ্যে মানুষের সভ্যতার ক্রমবর্দ্ধমান কাজের চাপের একটা চমৎকার ইতিহাস রয়েছে — এগুলো ঠিক এরকমই হয়েছে কারণ একটু একটু করে বদলাতে থাকা কাজের প্রয়োজন আর সেই কাজ সামলে ওঠার মানুষের সামর্থ্য — দুটোই একইসঙ্গে নিরবচ্ছিন্ন ভাবে বদলাতে থেকেছে, থাকছে। ইতিহাস মানে একদম কাজের নিরিখে কাজের প্রয়োজন আর সামর্থ — এই দুজনের একত্র ইতিহাস।

।। দিন তিন ।।

১ ।। কম্পিউটার হার্ডওয়ারের প্রাগ-ইতিহাস

আমাদের এই গ্নু-লিনাক্স ইশকুল পাঠমালার পরিকল্পিত ছকটা শুধুই বদলে যাচ্ছে। শেষ বদলটা তো ঠ্যাংজনিত। প্রথম ছকে তিন যেটা লিখেছিলাম, এই নতুন খসড়ায় এমনিতেই সেটা বদলে যেত কারণ শূন্যটা আগে ছিলনা, এখন এসেছে। আর, আগের খসড়াতেও পরে বেশ মন খুঁত খুঁত করেছে — শুধু কম্পিউটারের যান্ত্রিক ইতিহাসটাই এসেছিল, সার্কিট আর চিপ আর কার্ডের পিছনে জ্যান্ত মননের বদলের ইতিহাসটা প্রায় ধরাই ছিলনা। এবার সেটা আনতে চাইছি। প্রথমে হার্ডওয়ার স্তরে বদলের ইতিহাসটা আমরা ধরব, তারপর চিন্তাপ্রক্রিয়ার স্তরে। সফটওয়ার বলতে পারতাম, কিন্তু কম্পিউটার শাস্ত্রে সফটওয়ার কথাটার একটা বিশেষিত টেকনিকাল অর্থ আছে, সফটওয়ার-ও চিন্তাপ্রক্রিয়া, আমাদের চিন্তাপ্রক্রিয়াকে নিজেদের মাথা থেকে বার করে কম্পিউটারে ভরে দেওয়া, যাতে ক্যালকুলেটরের মত সবকিছুই নিজের হাতে নিজের মাথায় না করতে হয়, কিছুটা সে নিজেই করে নিতে পারে,

প্রোগ্রামিং ভাষার ব্যাকরণের ছাচে ঢালা আমাদের ওই বকলমা চিন্তাপ্রক্রিয়াকে অনুসরণ করে। তাই, বরং সাইবার-মনন বলাই ভালো।

যন্ত্র আর মনন এই দুটো ইতিহাসের উপাদানগুলো মিলে যাওয়া দরকার, যাতে এর পরের স্তরে আমাদের সিস্টেম বোঝার চেষ্টাগুলো অসার না হয়ে পড়ে, কারণ, গ্নু-লিনাক্স অপারেটিং সিস্টেমের কার্যকারণ সম্পর্কগুলো বহু জায়গাতেই গড়ে উঠেছে এই ইতিহাসটার সাপেক্ষে। আর মিশেল ফুকোর অর্ডার অফ ডিসকোর্সের সেই ঘোষণাটা তো আছেই — আসলে আমাদের প্রতিটি আলোচনাই সমালোচনা, আগের ঘটে যাওয়া আলোচনাগুলোর দুলতে থাকা লেজের পিছনে ব্যাদিত দাঁতের প্রদর্শনী। তাই পুরোনোটা আছেই, আমরা খেয়াল করি আর না-করি। আর, খেয়াল করাটা বহুসময়ই বেধড়ক জরুরি হয়ে পড়ে, একটা অপারেটিং সিস্টেমকে তার গতিবিজ্ঞানের নিরিখে ধরতে গেলে। অপারেটিং সিস্টেমটা শুধু জানা নয়, ভাবতে পারা দরকার।

আমরা শুরু করব যন্ত্রের বদলের প্রাগ-ইতিহাস দিয়ে। প্রাগ-ইতিহাস, কারণ, আমরা যাকে কম্পিউটার বলে চিনি, সেই রকম বা তার কাছাকাছি হার্ডওয়ার এবং সফটওয়ার সংস্থানটা আসতে শুরু করার পর থেকে আমরা তাকে দেখতে আরম্ভ করব অপারেটিং সিস্টেমের নিরিখে। মানে চার নম্বর দিন থেকে। এই জায়গাটা খেয়াল করুন, একদম গোড়ার দিকে, কম্পিউটারের অপারেশন আর অপারেটিং সিস্টেম কিন্তু আলাদা হয়ে উঠতে পারেনি। একটা কাজ করা মানে ছিল গোটা আদেশটা সরাসরি কম্পিউটারকে তার একদম হার্ডওয়ারের বোধ্য রকমে গুঁজে দেওয়া। পরে, কাজের জটিলতাটা যত বাড়তে লাগল, তত আরো আরো বেশি পরিমাণে আরো আরো জটিল কাজ করার দরকার পড়ল, তখন, দেখা গেল, কাজের ক্ষেত্রটা প্রস্তুত রাখার দায়িত্ব দেওয়া দরকার পড়ল স্বতন্ত্র একটা ব্যবস্থাকে। কাজের এই জমিটা তৈরি রাখে অপারেটিং সিস্টেম, তাতে বীজ পোতা আর ফসল কাটার কাজ করে সফটওয়ার। ক্রমে ক্রমে আরো স্পষ্ট হবে ব্যাপারটা এই ইতিহাসের আলোচনাটা চলতে চলতে, তিনে, প্রাক-ইতিহাসে, আর চারে, জেনারেশন থেকে জেনারেশনে কম্পিউটারের ইতিহাসে।

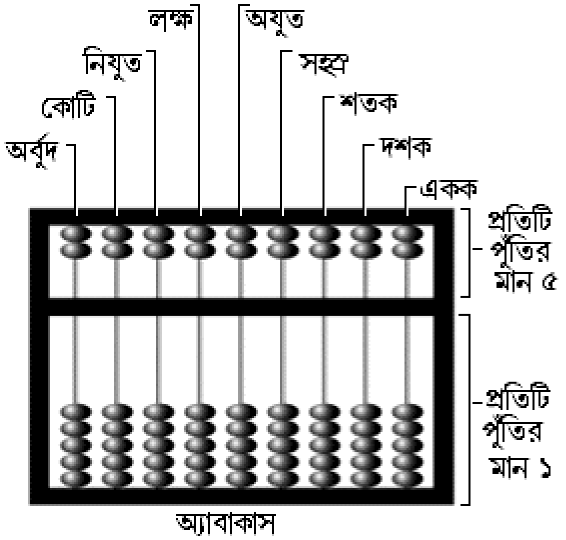
এই জেনারেশনের ইতিহাসটা শুরু হয় দ্বিতীয় বিশ্বযুদ্ধ নাগাদ একই সঙ্গে পৃথিবীর চার জায়গায় ডিজিটাল কম্পিউটার আবিষ্কারের কাজ শুরু হওয়ায়। চল্লিশের দশকের মাঝামাঝি নাগাদ, মূলত যুদ্ধবাজির সরকারি তাগিদেই, প্রায় একই সঙ্গে এই ভাবনাচিন্তা শুরু হয় আমেরিকার হার্ভার্ড প্রিন্সটন পেনসিলভেনিয়া বিশ্ববিদ্যালয়ে এবং জার্মানিতে। হার্ভার্ড বিশ্ববিদ্যালয়ে হাওয়ার্ড আইকেন, প্রিন্সটনে ইন্সটিটিউট ফর অ্যাডভান্সড স্টাডিজ-এ ভন নয়ম্যান, পেনসিলভ্যানিয়া বিশ্ববিদ্যালয়ে উইলিয়ম মশলে, এবং জার্মানির কনরাড জিউসে — এরা সবাইই এক একটা ডিজিটাল বা সংখ্যাভিত্তিক হিশেব-যন্ত্র খাড়া করেন। কিন্তু এই সময়কার এই মেশিনগুলো দেখলে আমাদের কম্পিউটারের চেয়ে কারখানা বলেই বেশি মনে হবে, আমাদের পরিচিতি পিসি-অবয়বের থেকে এরা এতটাই আলাদা। আমাদের পরিচিত ধাঁচের পিসির ধারণাটাই আসতে শুরু করেছিল, বলা যায়, ডেক বা ডিইসির পিডিপি সিরিজের সময় থেকে, তার মানে একষট্টির আগে নয়। এই পিডিপি সিরিজের মিনি-কম্পিউটারের কল্যানেই আমরা পেয়েছিলাম আজকের কম্পিউটার জগতের সবচেয়ে বড় দুটো চালিকাশক্তি ইউনিক্স এবং সি। যাকগে সে অনেক পরের কথা, এখন এসবে আমাদের দরকার নেই, আমরা এখন যেতে চাইছি কম্পিউটার যন্ত্র এবং মননের প্রাকইতিহাসে।

এই প্রাকইতিহাসের কতকগুলো বিন্দু আমাদের কাছে খুবই পরিচিত এবং সার্বজনীন, যেমন ব্যাবেজের ডিফারেন্স ইঞ্জিন বা অ্যানালিটিকাল ইঞ্জিন, বহু জায়গাতেই জনপ্রিয় আলোচনায় প্রায়ই দেখি এদের উল্লেখ। কিন্তু অন্য অনেকগুলো বিন্দুই তা নয়। যেমন সবচেয়ে বড় প্রশ্নটা হল, ঠিক কোথা থেকে শুরু করব? প্রথমে ভেবেছিলাম, সিন্ধু সভ্যতার নিজস্ব ‘হরপ্পা ফুট’ এই পরিমাপ থেকে, বা তৈত্তিরিয় সংহিতা বা শতপথ ব্রাহ্মণের গণিত থেকে, বা শূল্বসূত্র, সেইজন্যে এই ঠ্যাং অ্যাকাউন্টে পড়ে-পাওয়া-আঠাশ-আনা ছুটির সাড়ে তিন আনা খরচ করলাম, নেটে একটা চমৎকার হিস্ট্রি অফ ম্যাথামেটিক্স পাওয়া যায়, নিজের পড়ার জন্যে নামিয়ে পিডিএফ করা ছিল, তারই সাইজ প্রায় আড়াই এমবি, সাইজ ছোট করতে গিয়ে সব লিংকও উড়িয়ে দিয়েছিলাম, কোন সাইট তাই আর মনে নেই, যতদূর মনে হচ্ছে স্ট্যানফোর্ড ইউনিভার্সিটি। সেই পিডিএফ থেকে অনেকটা পড়লাম, কিন্তু শেষ অব্দি সেখান থেকে শুরু করিনি আর, লিখতে হয়ত ভালো লাগত, কিন্তু আমাদের কাজের জন্যে জরুরি হত না। শুরু করেছি অ্যাবাকাস থেকে, আপনাদের কথা ভেবে, আপনাদের ঠ্যাং আস্ত আছে, বই পড়ার ছুটি তো পাবেন না।

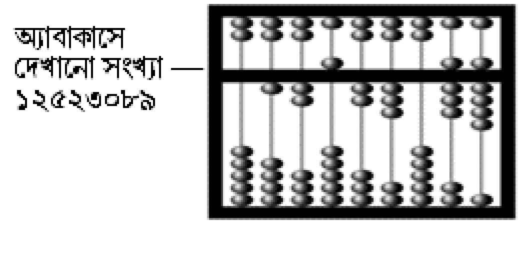
আগেই বলেছি, আজকের কম্পিউটার ডিজিটাল কম্পিউটার, ডিজিট মানে অঙ্কই তার ইনপুট, অঙ্কই তার আউটপুট, তাই এর ইতিহাসেরও শুরু সেই সময়টায়, অঙ্ক কাম সংখ্যাকে নাড়াচাড়া করতে করতে মানুষ যখন মনে করল, এর জন্যে একটা যন্ত্র হলে ভালো হত। অ্যাবাকাস থেকে এর শুরু। আবার, আপনি যেখানে এই লেখাটা পড়ছেন সেই পিসিটা তার আর একটা ধারা, ক্যালকুলেটর আর একটা, এরকম আরো আছে, কিছু ধারা হারিয়েও গেছে।

আমাদের ইতিহাসে প্রাচীনতম সংখ্যার উদাহরণ মূলত এসেছে মেসোপটেমিয়া থেকে। মেসোপটেমিয়ার মাটির ফলকে প্রচুর সংখ্যা এবং তাদের পাটীগণিতের উদাহরণ পাওয়া যায়। এগুলো প্রায় সাড়ে চার হাজার বছরের পুরোনো। এবং এদের অনেকটাই হল হিশেব এবং ট্যাক্সের অঙ্ক। এই ট্যাক্সের ব্যাপারটা সিন্ধু উপত্যকার পাথরের ফলকগুলোর বেলাতেও সত্যি। আস্কো পার্পোলা তার 'ডিসাইফারিং ইন্ডাস স্ক্রিপ্টস' বইতে যেভাবে লিখেছেন, ঠিক সবসময়ই ঠিক ট্যাক্স না হলেও টাকাপয়সার হিশেব রাখার প্রয়োজন আমাদের অঙ্কবিবর্তনের সবচেয়ে শক্তিশালী প্যালা। গ্রীকরা সংখ্যা লিখত অক্ষর দিয়ে, রোমানরা আরো খারাপ। ভারতীয় গণিতের অবস্থাটা ছিল এদের চেয়ে একটু বেটার জায়গায়। শূন্য নম্বর দিনে আমরা যে ডেসিমাল বা দশমিক পদ্ধতির কথা আলোচনা করলাম সেটা ভারতীয় গণিতের অবদান, দশমিক ব্যবস্থার এই অবস্থানের ম্যাজিক — একক দশক শতক সহস্র ইত্যাদি প্রত্যেকটা অবস্থানে দশের শক্তি ১ করে বেড়ে যাওয়া। এই জায়গাটায় আমরা এখনো পশ্চিমের চেয়ে এগিয়ে আছি। যেমন ধরুন তেতাল্লিশ। ইংরেজরা বলে ফর্টি থ্রি। অথচ আমরা বলি তেতাল্লিশ মানে তিন চল্লিশ। আগে একক বলছি, তার পর দশকের অঙ্ক। গণিতের দিক দিয়ে বেশি সঠিক।

ভারতীয় গণিতের আর একটা অবদান শূন্য। কিন্তু শূন্যর ইতিহাস নিয়ে অনেক জটিলতা আছে। শূন্য মানে কোনটাকে বুঝব? একটা তার আঙ্কিক মান, অন্যটা একটা অবস্থানগত চিহ্ন। শূন্য হল তাই যার কোনো মান নেই, না ধনাত্মক না ঋণাত্মক, পাঁচ থেকে পাঁচ বিয়োগ করে যা পাই — এই মানটা বুঝব? নাকি, অবস্থানের চিহ্নটাকে? মানে, ২০১৫ কে যা ২১৫ থেকে পৃথক করে সেটাও তো শূন্য। এই জটিলতা সত্ত্বেও, বোধহয়, সেই দুই অর্থেই, শূন্যটা ভারতীয় গণিতেরই অবদান। যাই হোক, ভারতীয় পারসিক রোমান গ্রীক যে বর্ণমালায় যে চিহ্নেই যে ভাবেই তাদের লেখা হোক, মানুষের জীবন ও সভ্যতা টিকে থেকেছে এই সংখ্যাদের হাতে নিয়েই। হাতে নিতে নিতে এত ভার লাগছিল যে ক্রমে ক্রমে তার সংখ্যা দেখাশুনোর একটা সহকারী রাখার কথা মনে হল। নানা দেশে নানা ভাবে এসেছে এই সহকারী। এর একটা ভালো উদাহরণ অ্যাবাকাস।



অ্যাবাকাস



১.১।। অ্যাবাকাস

বাচ্চা বয়সে আমাদের হাতে যে স্লেট আসত, তার একটু ডিলাক্স এডিশনগুলোয় একটা করে অ্যাবাকাসও লাগানো থাকত। আমার নিজের মনে আছে, অনেকবার ভেবেছি, এটা কী করে, কিন্তু হাতের কাছে কেউই সেটা জানে এমন পাইনি। যেরকম অনেক কৌতূহলই মেটেনা আমাদের। ইন ফ্যাক্ট সঠিক ভাবে হিশেব করার প্রক্রিয়াটা আমি জানলাম এই পাঠমালাটা লিখতে শুরু করে, অ্যাবাকাস ব্যাপারটা বোঝার চেষ্টায়। বেশ মজার লাগল, তাই ছবিটা দিলাম, সঙ্গে এর সরল নিয়ম কটাও। গুটি বা পুঁতিগুলো নেড়ে দেখুন, মানে, মনে মনে, বেশ ইন্টারেস্টিং। কিন্তু, যোগ এবং বিয়োগের ব্যাপারে ভালো কাজ দিলেও, গুণ বা ভাগের বেলায় অ্যাবাকাস বেশ অচল। যদি সংখ্যাটা খুব বড় হয় তাহলে হিশেব করাটা রীতিমত অসম্ভব হয়ে পড়ে। কিন্তু গুণ বা ভাগোপযোগী উপযুক্ততর খুড়োর কল আসতে তখনো অনেকটা দেরি।

যেহেতু চীন বা জাপান ছাড়া আজকে আর কোথাও বোধহয় অ্যাবাকাসের জ্যান্ত ব্যবহার নেই, সেইজন্যেই হয়ত আমাদের মাথায় অ্যাবাকাস বললেই চৈনিক শব্দটা আসে। কিন্তু তার আর পুঁতি দিয়ে বানানো এই গোনার যন্ত্রটা, নানা চেহারায়, ব্যবহার হত প্রায় সবকটা প্রাচীন সভ্যতাতেই। অ্যাবাকাস শব্দটা এসেছে গ্রীক অ্যাবাক্স থেকে, মানে গোনার বোর্ড বা চৌকো ভূমি, সেটা আবার এসেছে হিব্রু অ্যাবাক থেকে — ধুলো বা জং এইসব। ধুলোয় পড়ে থাকতে থাকতে অ্যাবাকাসে জং পড়ে যেত? এত কম হিশেব করতে লোকে? এত কম ট্যাক্স দিত লোকে? এত ভালো ছিল রাজারা?

ঠিক আজকের ক্যালকুলেটরের মতই, হিশেব করার কাজ অ্যাবাকাসে বেড়ে চলত বটে, কিন্তু, শরীর শরীর তুমি শুধুই শরীর অ্যাবাকাস, তোমার মন কই? কোনো নিয়ন্ত্রক সংস্থা তো ছিলই না, শূন্য নম্বর দিনে ক্যালকুলেটরেরও যে সমস্যার কথা আমরা বলেছি, এমনকি ক্যালকুলেটর তো সংখ্যাগুলোকে বোঝে বিমূর্ত চিহ্ন দিয়ে, ১, ২, ৩, ... ইত্যাদি, অ্যাবাকাস আরো নিরেট, সংখ্যাগুলোকেও বোঝাত কোনো বিমূর্ত চিহ্ন নয়, গুটি বা পুঁতি দিয়ে। গুটি বা পুঁতির সংখ্যাই হল হিশেবের সংখ্যা।

ছবিটায় দেখুন, যতদূর সম্ভব মধ্যযুগের একটা উডকাট ছবি, ইউরোপের। গ্রেগর রাইশের 'মার্গারিটা ফিলোসফিকা' বলে একটা বই থেকে। ছবিটার মধ্যে একই সাথে দুটো ঘটনা ঘটছে। বাঁ দিকে বিমূর্ত অক্ষরে সংখ্যায় হিশেব করছে একজন, আর ডানদিকে, মূর্ত রকমে, একটা বোর্ডের উপর নুড়ি পাথর দিয়ে আর একজন করছে সেই একই হিশেব। বোধহয় এইরকম জং পড়া তারের ছক কাটা ধুলোধূসর বোর্ড থেকেই অ্যাবাকাসের নামটা এসে থাকবে। ডানদিকের ব্যাপারটার সঙ্গে অ্যাবাকাসের গঠনগত মিলটা দেখুন।

ছবির মূল টেনশনটা খেয়াল করুন — বাঁদিক আর ডানদিক, সংখ্যা আর নুড়ি, বিমূর্ত আর মূর্ত। যে মানুষ কেবল অঙ্ক আর সংখ্যা চিনতে শিখেছিল তার কাছে সংখ্যাগুলো ছিল বাস্তব মূর্ত অবয়বী। এই আছে দুটো চমরী গাই, আর দুটো মারলাম, হল চারটে। সংখ্যা তখনো মূর্ততার লেজুড়। কিন্তু বিদ্যা আর বিজ্ঞান শুরু হতে হতে সংখ্যা যথেষ্ট বিমূর্ত

হয়ে পড়বে, ক্লাসিকাল ইউরোপের এই ছবি সেই গতিটাকেই দেখাচ্ছে। একটু সংখ্যা আর অঙ্ক চিনে যাওয়া মানুষ যখন ঋণাত্মক সংখ্যা আর ২-এর বর্গমূল বার করতে শুরু করবে, তখন তাতে আর মূর্ততা কই। ভেবে দেখুন, ঋণাত্মক কোনো চমরী গাই দেখেছেন কখনো? আমি অবশ্য কোনো চমরী গাই-ই কখনো দেখিনি। উদাহরণটা মাথায় এলো কারণ আমার শিশুবয়সে আদিম মানুষদের নিয়ে যে বইটা দেওয়া হয়েছিল পড়তে, তাতে তারা শুধু চমরী গাই-ই শিকার করত। অথবা, ২-এর বর্গমূলটা ধরুন, এটাকে আপনি দশমিকের পর দশ বিশ পঞ্চাশ যতদূর অব্দি চান, হিশেব করে যেতে পারেন, কিন্তু আপনার ফিতে বা স্কেল দিয়ে তার ধারেকাছে কিছুও কোনোদিন মেপে উঠতে পারবেন? এটাই হল গণিতের বিমূর্ত যাত্রার বাড়তি শক্তি। এই ছবিটা যে আমলে আঁকা তখন গোটা পৃথিবীর নানা জায়গাতেই অ্যাবাকাস একটা পরিচিত জিনিষ। ছবিটা দেখে আর একটা কারণেও আমি মজা পাচ্ছি। গণিতে কলনবিদ্যা বা ক্যালকুলাস আর আমাদের রোজকার হিশেব করা বা ক্যালকুলেট শব্দটা এসেছে ল্যাটিন ক্যালকুলাস থেকে, যার মানে ছোট পাথর। প্রথম যেদিন এটা জানলাম সেদিন বুঝেছিলাম পরশুরামের চিকিৎসা সঙ্কটে, 'তোমার পেটে ক্যালকুলাস হয়েছে' — এই বাক্যটার মানে কী।

মেসোপটেমিয়া বা মিশরের পাটীগণিত ইতিহাসের একটা স্তর জুড়ে আছে। এই প্রাচীন স্তরেই আছে ভারতবর্ষের ওই শূন্য তথা পরবর্তী গণিত, তারপরে কিছুটা আরো এগিয়েছিল জৈন গণিতচর্চা, বাখশালি পুঁথিতে যার নিদর্শন আছে। এরপর একটা দীর্ঘ সময় জুড়ে একটা অন্ধকার যুগ চলে, যার পরে গণিতকে আমরা ফের আবিষ্কার করি, এবার ইউরোপে, ইউরোপের ইতিহাসে যেটাকে ক্লাসিকাল যুগ বলে, সেই সময়ে।

এই দুটো স্তরের মধ্যে কিছু যে হয়নি একদম তা নয়। মূলত আরব গণিতবিদরা, যতদূর সম্ভব ভারত থেকে তাদের গণিতচর্চাটা পান, টিকিয়ে রেখেছিলেন ভারতীয় বীজগণিত, পাটীগণিত এবং জ্যামিতির চর্চা। এই আরব গণিতবিদরা শুধু ভারতীয় নয়, গ্রীক গণিতের ঘরানাটাও পেয়েছিলেন। এদের একজন, মহম্মদ ইবন মুসা আল খোয়ারিজমি, লিখেছিলেন ভারতীয় সংখ্যাতত্ত্ব নিয়ে একটা বই, তার নাম 'ইলম আল-জাবর ওয়াল মুকাবলা', রিডাকশন এবং ক্যানসেলেশন, ছোট করা এবং কেটে দেওয়ার বিজ্ঞান, যার 'আল-জাবর' থেকে আলজেব্রা এসেছে। এবং আজকের গণিতের তথা কম্পিউটার শাস্ত্রের ব্যাপকতম ব্যবহৃত আর একটা শব্দ এসেছে ওই লেখকের নাম থেকে — আলখোয়ারিজমি থেকে আলগোরিজমি, তার ল্যাটিন সংস্করণ থেকে এসেছে অ্যালগরিদম শব্দটা। যার খুব কাছাকাছি 'ক্রিয়াপদ্ধতি' ছাড়া আর কিছু এখন লিখতে গিয়ে মাথায় আসছে না। আমাদের এই পাঠমালাতেও এই ক্রিয়াপদ্ধতি বা অ্যালগরিদম বারবার আসবে।

আরব থেকে এই গণিতচর্চা খুব দ্রুত চলে গেছিল ইউরোপে। ইউরোপ তখন প্রচন্ড গতিতে তার নিজের বাইরে বেড়ে উঠতে চাইছে। খাই খাই চলছে, আহারে বসতে আরো কিছু দেরি আছে। আল-খোয়ারিজমির বই ল্যাটিনে অনুবাদ হয় দ্বাদশ শতাব্দীতে। ঠিক যে শতাব্দীতেই আবিষ্কৃত হয়েছিল কম্পাস — ভূ পর্যটন। যদিও ঠিক ভাবে স্টার্ট নিতে আরো কিছুটা সময় লাগবে। মূল গতিটা শুরু হবে আরো বেশ কিছু দিন পরে, ১৪৭৭-এ, যে বছর ভারতযাত্রার টাকা তোলার আশায় ইংলন্ডে এলেন কলম্বাস। সেই একই বছরে ইউরোপে স্থাপিত হল তিনটে বিশ্ববিদ্যালয় — সুইডেনের উপসালা, জার্মানির টুবিংগেন আর মেইনজ। এরও সতেরো বছর বাদে জেনোয়ার বেনে সান্তো স্তেফানো আদা আর গোলমরিচ কিনতে জাহাজে করে কালিকট আসবেন ১৪৯৪-এ। এর আগেই, ১৮৯২-এ, শুক্রবার ৩ অগাস্ট ক্রিস্টোফার কলম্বাস যাত্রা করলেন নিনা পিন্টা এবং সান্টামারিয়া এই তিন জাহাজ নিয়ে, ডাঙা দেখতে পেলেন ১২ অক্টোবর। পৃথিবী, জ্যোতির্বিদ্যা, মাপজোক — এর সঙ্গে সঙ্গে তাল মিলিয়ে আর একটা জিনিষ আসার কথা, নিজের মাথাতেই ভেবে দেখুন — সময়, সময়-ধারণা, এবং সময়ের নিখুঁত মাপজোক। ঠিক তাই হল, শুধু গণিত নয়, আর একটা জিনিষ নাটকীয় ভাবে বদলাচ্ছিল এই গোটা সময়টা ধরে, সেটা হল সময়, এবং তার সঙ্গে মানুষের সম্পর্ক। ১৫১১-য় জার্মানির ন্যুরেমবার্গ ক্রনিকলসে পৃথিবীর প্রথম, মিনিটের কাঁটা ছাড়া শুধু ঘন্টার কাঁটায়, ঘড়ির বিজ্ঞাপন বেরোবে। জার্মান গণিতবিদ জেমা ফ্রিসিয়াস গোটা পৃথিবীর জন্যে একই সময়-কাঠামো, মানে সঠিক অর্থে গ্লোবালাইজেন শুরু করার গ্রিনউইচ-মিন-টাইম, আন্তর্জাতিক-সময়-ধারণার ভ্রূণ রাখবেন ১৫৩০-এ। কলোনি বিপ্লব আসছে, চলো যোগ দিয়ে আসি, ইউরোপ জুড়ে তখন সাজো সাজো রব। তারও তেরো বছর বাদে, ১৫৪৩-এ, মিখোলেই কোপেরনিক নামে এক প্রুশ-পোলিশ, পৃথিবীর সবকিছুই বদলে নেওয়ার অধিকার তাদের আছে এই বিশ্বাস থেকেই ইংরেজরা তার নাম বদলে করেছিল নিকোলাস কোপার্নিকাস, লিখবেন 'ডে রেভলিউশনিবাস অর্বিয়াম

কোয়েলেস্টিয়াম', তিনি যার এক নম্বর খসড়া 'কমেন্টারিওলাস' লিখেছিলেন ১৫০৭ থেকে ১৫১৫-র মধ্যে, যাতে প্রথম রেভলিউশন শব্দটা ব্যবহার হবে — সম্বর্তন বিবর্তন আবর্তন আদি। ১৫৫০-এ, ওই কোপারনিকাসের ছাত্র জোয়াচিম ভন লাউচেন ছাপবেন, যতদূর সম্ভব, পৃথিবীর প্রথম ত্রিকোণমিতির মানের তালিকা, ট্রিগোনোমেট্রিক টেবল, পরবর্তী কালের জ্যামিতির বিকাশকে যা ব্যাপকভাবে প্রভাবিত করবে। যার বিকাশের চূড়ান্ত জায়গা দাঁড়াবে, আরো বেশ কিছু বছর পরে, কার্টেজিয়ান কো-অর্ডিনেট, দেকার্তের স্থানাঙ্ক জ্যামিতি, ১৬৩৭। গণিতচর্চার একটা বড় উৎসাহ এসেছিল এই উপনিবেশ বানানোর তাগিদ থেকে। নতুন জংলি দেশ খুঁজতে হবে, তার মানে জাহাজ নিয়ে সমুদ্রে, তার পথ, তার ম্যাপ, তার মাপ।

বেড়ে উঠেছিল সমুদ্রযাত্রা, তাই বাধ্যতামূলক ভাবেই গণিত আর জ্যোতির্বিদ্যা। তার মানেই আরো আরো হিশেব। পরিমাপের পদ্ধতিগুলোও বদলাচ্ছিল। ওই ফিতেয় আর কাজ চলবেনা যাতে একটাই মাপ লেখা — ছাব্বিশ, তাই গলা পেট কোমর সবই ছাব্বিশ, জাতি-কাল-দেশ নির্বিশেষে সবাইকে সব-কিছুকে যা শুয়োর করে দেয়। তখনকার লোক কিছু হিশেব করতে পারত-ও বটে, হিশেব করার পেশি থাকত আলাদা। জ্যোতির্বিদ কেপলারকে ভাবুন, কুড়ি বছর একটানা শুধু হিশেবই করে গেছেন। যোগ বিয়োগ গুণ আর ভাগ করার, বর্গমূল ঘনমূল মানে স্কোয়ার বা কিউব রুট ইত্যাদি করার, এবং ক্রমে আরো আরো বড় বড় সংখ্যা দিয়ে করার, প্রয়োজনটা রোজই বাড়ছিল।

১.২।। লগারিদম ও স্লাইড রুল

এখানে একটা নাটকীয় বদল এনেছিল স্কটল্যান্ডের গণিতবিদ জন নাপিয়ের-এর লগারিদম। এই জন নাপিয়েরের

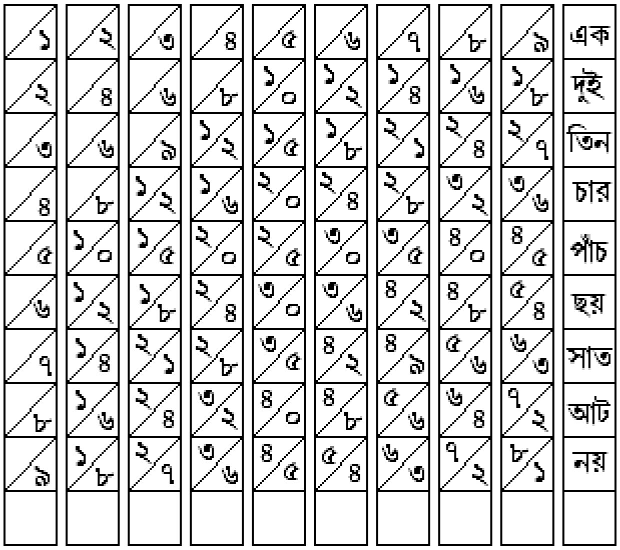
জন নাপিয়ের

(১৫৫০-১৬১৭) গণিতটা ছিল সময় কাটানোর খেলা। নাপিয়ের নিজের প্রাথমিক এবং অবশ্যই গুরুত্বপূর্ণতর কাজ বলে মনে করতেন বাইবেলের রিভিলেশন তত্ত্ব নিয়ে তার গবেষণাকে। এই কাজে তিনি যথেষ্ট সফল হয়েছেন বলেও মনে করতেন। আজ তার এই বাইবেল গবেষণাকে জানা যেতে পারে একমাত্র নাপিয়েরকে নিয়ে গবেষণা করলেই। অথচ তাঁর এই লগের গুতোয় গোটা ধরিত্রী জুড়ে গুণ হয়ে দাঁড়াল যোগ আর ভাগ মানে বিয়োগ।

আমাদের কাছে জন নাপিয়েরই বেশি পরিচিত, কিন্তু নাপিয়েরের সঙ্গে প্রায় একই সময়ে, মাত্র চার বছর পরে, স্বতন্ত্র এবং স্বাধীনভাবে লগ-টেবিল ছাপিয়েছিলেন আর একজন সুইৎজারল্যান্ডবাসী গণিতবিদ, তার নাম জাস্টাস বিরগিয়াস (১৫৫২-১৬৩২), তাঁর লগ-টেবিল ছাপা হয় ১৬২০-তে। আমরা আজ লগকে যে চেহারায় চিনি সেই চেহারায় নিয়ে আসেন এক ইংরেজ গণিতবিদ, হেনরি ব্রিগস (১৫৬১-১৬৩০)। লগারিদম শব্দটাও তৈরি নাপিয়ের-এর। গ্রীক লোগোস — অনুপাত বা হার, এবং অ্যারিথমোস — সংখ্যা। শব্দটা আসার আগে নাপিয়ের এদের ডাকতেন কৃত্রিম সংখ্যা বলে। লগারিদম তালিকা তৈরি করতে নাপিয়ের-এর কুড়ি বছর লেগেছিল। আগেই বললাম, এটা আমাদের কাছে লগারিদম-এর পরিচিত অবয়ব নয়, আর নাপিয়েরিয়ান লগ বলে এখন যেটা চলে, নাপিয়েরের এই লগ কিন্তু সেটাও নয়।

পরবর্তী জীবনে নাপিয়ের আর একটা জিনিষ তৈরি করেছিলেন, অ্যাবাকাসের থেকে একটু উন্নত এক ধরনের ক্যালকুলেটর। মান্ধাতার বাবার মানে যুবনাশ্বের ক্যালকুলেটর বলা যায়। এর নাম 'নাপিয়েরস বোনস' মানে 'নাপিয়েরের হাড়', সেটা তৈরি হত, না না, নাপিয়েরের নয়, একটা মানুষের আর কত হাড় হয়, হাতির দাঁত বা অন্য প্রাণীর হাড় থেকে।

নাপিয়েরের হাড়

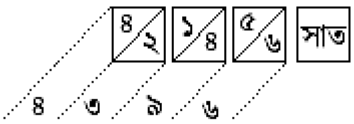


সঙ্গের ছবিতে দেখুন, এই সেই নাপিয়েরের হাড়, যুবনাশ্বের ক্যালকুলেটর। বেশ মজার, একটু নেড়ে দেখুন। আসুন, যুবনাশ্ব এই ক্যালকুলেটর ব্যবহার করে কতজন শত্রুকে প্যাঁদালেন তার হিশেব কী করে করতেন, সেটা একটু দেখি। এই দেখাটা সম্পূর্ণ অপ্রয়োজনীয়, কিন্তু বেশ মজার, লিনাক্স হাউ-টু-র ভাষায়, একসেলেন্ট মিস-ইউজ অফ রিসোর্স। যান্ত্রিক ক্যালকুলেটরের বিবর্তনের ইতিহাসে ঠিক এর পরের স্তরে শিকার্ডের মেশিন আবার এই নাপিয়েরের হাড়কেই ব্যবহার করেছিল তার গঠনে। এই এক দুই তিন লেখা ডানদিকের হাড়টা, এখানে বাংলায়, লেখা হত রোমান-এ, আর সংখ্যাগুলো ল্যাটিন, মানে আমাদের অভ্যস্ত ইংরিজি অঙ্ক।

ধরুন, আমরা হিশেব করতে চাইছি, একটানা ৬২৮ দিন ধরে রোজ যুবনাশ্ব ৪২৭টা করে শত্রু সংহার করে থাকলে, মোট কত শত্রু-সংহার করেছেন। আজ, যুদ্ধ শুরুর প্রায় দুবছর বাদে, ক্লান্ত যুবনাশ্ব হিশেব করছেন, মোট কটা গারবেজ কমাতে পেরেছেন মা ধরিত্রীর বুক থেকে। তার মানে, তাকে গুণ করতে হবে ৬২৮-কে ৪২৭ দিয়ে।

আমরা তাই ৬ নম্বর ২ নম্বর আর ৮ নম্বর হাড়টা বার করে আনব, ডানদিকের স্থানচিহ্ন দেওয়া হাড়ের এক, দুই আর তিন নম্বর সারির সঙ্গে মিলিয়ে রাখব তাদের। এবার খেয়াল করুন, ৪২৭ দিয়ে গুণ করা মানে আসলে ৪ খানা একশো, ২ খানা দশ আর ৭ খানা এক দিয়ে গুণ করা। ছোটবেলায় যেভাবে একক দশক শতক দিয়ে ভাবতেন, শূন্য নম্বর দিনে আমরা যা আর একবার মকশো করে এলাম, সেই ভাবে ভাবুন।

আমরা শুরু করব ৭ খানা এক দিয়ে গুণ করে। সাত নম্বর সারিটা দেখুন। এর সঙ্গে মিলিয়ে কোনাকুনি আমরা ফল গুলো লিখে ফেললাম।



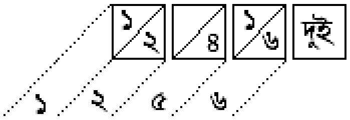
৬ = ৬

৫ + ৪ = ৯

১ + ২ = ৩

৪ = ৪

তার মানে একের সারির গুণফল ৪৩৯৬। এবার গুণ করব ওই ২ খানা দশ দিয়ে। দ্বিতীয় সারিটা দেখুন এবার। এবং একই ভাবে ফলাফলটা লিখে ফেলা যাক।



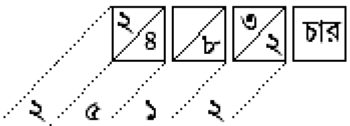
৬ = ৬

১ + ৪ = ৫

২ = ২

১ = ১

তার মানে, দশের সারির গুণফল হল ১২৫৬ গুণ দশ মানে ১২৫৬০। দশ দিয়ে গুণ করলাম কারণ এটা দশের সারি। এবার একের সারি আর দশের সারি হয়ে গেছে। বাকি আছে একশোর সারি। একই ভাবে আমরা ৪ খানা একশো দিয়ে গুণ করব। চার-নম্বর সারিটাকে দেখুন। এই বারে একটু জটিলতা আছে হাতে থাকার ব্যাপারটায়।



২ = ২

৮ + ৩ = ১ (হাতে রাখো ১)

৪ + ১ = ৫

২ = ২

এবার এই গুণফল ২৫১২ কে একশো দিয়ে গুণ করে পাই ২৫১২০০, এটাই আমাদের এই একশোর সারির গুণফল। এবার মোট গুণফল পাব এই তিনটে সারির গুণফলকে যোগ করে।

৪৩৯৬ + ১২৫৬০ + ২৫১২০০ = ২৬৮১৫৬

এই উদাহরণের চেয়ে অনেক বেশি তাড়াতাড়ি এই গুণটা করা যাবে, একটু অভ্যস্ত হয়ে গেলেই। যাকগে, এই হল যুবনাশ্বের ক্যালকুলেটর, নেপিয়ারের হাড়, নিজের ডিজিটাল ক্যালকুলেটরের বা কম্পিউটারের সঙ্গে একবার তুলনা করুন, তাহলেই বুঝবেন আজকের মেশিন কতটা পথ পেরিয়ে এসেছে শ পাঁচেক বছরে।

লগারিদম ল্যান্ড করে যাওয়ার পরে প্রায় চারশো বছর ধরে, ইলেকট্রনিক ক্যালকুলেটর এবং পরে কম্পিউটার আসার আগে অব্দি গোটা পৃথিবীর মোট হিশেবনিকেশ চলেছে এই লগ টেবিলের দৌলতে। আমরাই ইশকুলে পড়াকালীন লগটেবিল কিনেছি, বইয়ের পিছনেও দেওয়া থাকত। লগটেবিল ধরে যেটা নিজে হাতে করতে হয়, সেই কাজটাই যান্ত্রিকভাবে করে দেওয়ার জন্যে এরপর এসেছিল স্লাইড রুল। স্লাইড রুল মানে দুটো পাশাপাশি লাগানো লগারিদমিক স্কেল যাতে গুণ ভাগ সহজে করে ফেলা যায় যোগ আর বিয়োগ করেই।

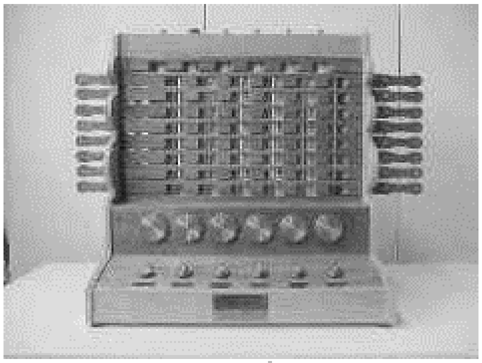
প্রথম সত্যিকারের স্লাইড রুল বানান ইংরেজ গণিতজ্ঞ উইলিয়াম উট্রেড(১৫৭৪-১৬৬০), ১৬২০-তে। এই উট্রেডই প্রথম গুণ এই অর্থে '×' চিহ্নটা ব্যবহার করেন। স্লাইড রুলকে বলা যায় অ্যানালগ ক্যালকুলেটর। অ্যানালগ যেখানে পরিমাপ দিয়ে হিশেব করে, ডিজিটাল করে সংখ্যা দিয়ে। ধরুন, ঘড়ির ডায়ালে মোট ৩৬০ ডিগ্রি কোণকে বারোটা ভাগে পরিমাপ করে এক একটাকে এক এক ঘন্টা দিয়ে বুঝছি, এটা অ্যানালগ। আর ওই সময়ই ডিজিট দিয়ে মানে অঙ্ক দিয়ে আসছে আমাদের ডিজিটাল হাতঘড়িতে। উট্রেড ছাড়া স্লাইড রুলের কাজ করেছেন এডমন্ড গুন্টের (১৬৮১-১৬২৫) এবং আমেডি ম্যানহেইম (১৮৩১-১৯০৬)। গুন্টের একটা দু ফুট লম্বা স্কেলের গায়ে লাগিয়ে দিয়েছিলেন একটা লগারিদমিক মানদণ্ড, যেখানে ডিভাইডর দিয়ে লগারিদমিক স্কেলের গায়ের মাপটা নিয়ে গুণফলটা

পেয়ে যাওয়া যেত। আমরা যে চেহারায় আধুনিক স্লাইড রুল দেখি, কাঁচের চিহ্নক বা কার্সর লাগানো — এই চেহারাটায় নিয়ে আসেন মানহেইম।

বিংশ শতাব্দীর গোড়া অব্দি ছিল এই লগটেবিল আর স্লাইড রুলের সর্বময় শাসন। টেবিলগুলোয় কাজ হত মোটামুটি নিখুঁত, কিন্তু হত বড় আস্তে। আর স্লাইড রুলে কাজটা দ্রুত হত ঠিকই কিন্তু সব উত্তরেই একটা 'প্রায়' মানে 'অ্যাপ্রক্সিমেশন' চিহ্ন (≅) জুড়তে হত। উনবিংশ শতাব্দীতে এসে আসতে শুরু করেছিল নানা ধরনের ক্যালকুলেটর, যার চূড়ান্ত আকারটাকে রূপ দিতে চেয়েছিলেন চার্লস ব্যাবেজ। কিন্তু একটা কথা, এই ক্যালকুলেটরগুলো প্রত্যেকটাই চলত যান্ত্রিক পদ্ধতিতে, হাতল, গিয়ার, প্যাঁচ ঘুরিয়ে। ইলেকট্রনিক তো দূরের কথা, ইলেকট্রিকও তখনো সিনে আসেনি।

১.৩।। স্লাইড রুল থেকে ব্যাবেজ

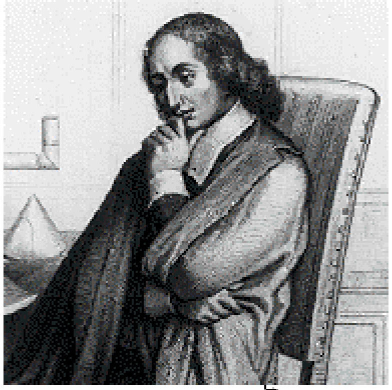
সত্যিকারের কম্পিউটার তৈরির বাস্তব জ্যান্ত প্রক্রিয়াটা শুরু হয় দ্বিতীয় বিশ্বযুদ্ধের আগে আগে, একই সঙ্গে পৃথিবীর অনেকগুলো জায়গায়। কিন্তু তার আগে অব্দি এই ধারার চিন্তাটা কিন্তু বন্ধ ছিলনা। আমরা ওই যুগের মেশিন বলতেই চার্লস ব্যাবেজকে বুঝি, কিন্তু সেখানে ব্যাবেজ একা এবং একলা নন। তা কখনো হয়না। একটা সময় তার নিজের গতিকে চারিয়ে দেয় মানুষের চিন্তায়, আলাদা আলাদা মানুষ আলাদা আলাদা জায়গায় একই দিকের ইঙ্গিত খুঁজতে থাকেন। সকলেই যে শেষ অব্দি একই ভাবে এগোতে পারেন তা নয়। ক্যালকুলাসের ইতিহাস ভাবুন, লিবনিতজ আর নিউটন — দুজনে মোটামুটি একই সময়ে আলাদা আলাদা ভাবে একই ক্যালকুলাসের দিকে এগিয়েছিলেন। বরং নিউটনের এগোনো নিয়ে তো অনেক সিরিয়াস প্রশ্ন উঠতে পারে, ব্রিফ হিস্ট্রি অফ টাইম-এর অ্যাপেন্ডিক্সে হকিং তুলেছেন। সেরকম কম্পিউটার নিয়ে ভাবনায় সকলেই ব্যাবেজের মত বেশ অনেকটা এগোতে পারেননি, কিন্তু চেষ্টা অনেকেই করেছেন, ব্যাবেজের আগে এবং পরে। তাদের অনেকের চেষ্টা প্রায় গোটাটাই নামের জগত থেকে হারিয়ে গেছে, কখনো কখনো রয়ে গেছে তাদের চারপাশের তাদের পরেকার অন্যদের ভাবনাচিন্তায়। এরকম কয়েকজনের উল্লেখ করা যাক।



শিকার্ডের ক্যালকুলেটিং ক্লক

১৬২৩-এ, জার্মান বৈজ্ঞানিক উইলহেল্ম শিকার্ড (১৫৯২-১৬৩৫) উদ্ভাবন করেন একটা যোগ, গুণ আর ভাগ করার মেশিন। তার নাম দেন ক্যালকুলেটিং ক্লক, হিশেব ঘড়ি। এখানে ১৬২৩-এর সেই ক্যালকুলেটিং ক্লক এর পুনর্নির্মাণের একটা ছবি দিলাম আমরা। যন্ত্রটা আসলে এক সেট নেপিয়ারের হাড়, এবং নিচে একটা গুনে যাওয়ার কাউন্টার। তার মানে যতটা যন্ত্র তার চেয়ে এটাকে একটা হ্যান্ডেল লাগানো নামতা বলাই ভালো।

উইলহেল্ম শিকার্ডের এই যন্ত্রটার কথা জানা গেছে একদম আধুনিক কালে। জ্যোতির্বিদ কেপলারের কাগজপত্র পড়তে গিয়ে। তাতে কেপলারকে লেখা শিকার্ডের একটা চিঠি পাওয়া যায়। চিঠিতে ছিল এই ক্যালকুলেটিং ক্লক নামের যন্ত্রটার বিবরণ আর একটা আলগা ছবি। এর পরে ১৯৫০ নাগাদ আরো কিছু নথীপত্র আবিষ্কার হয়, ক্যালকুলেটিং ক্লক সম্পর্কে আরো বহু খুঁটিনাটি জানতে পারা যায়। যা থেকে যন্ত্রটাকে ফের বানানো হয়েছিল। শিকার্ডের চিঠিতে আছে, মূল যন্ত্রটা অসমাপ্ত অবস্থাতেই আগুনে পুড়ে নষ্ট হয়ে যায়। পরে শিকার্ড আর নতুন করে কোনো ক্যালকুলেটিং ক্লক বানিয়েছিলেন কিনা তাও জানা যায়না। ১৬৩৫-এ প্লেগ রোগে মারা যান শিকার্ড।

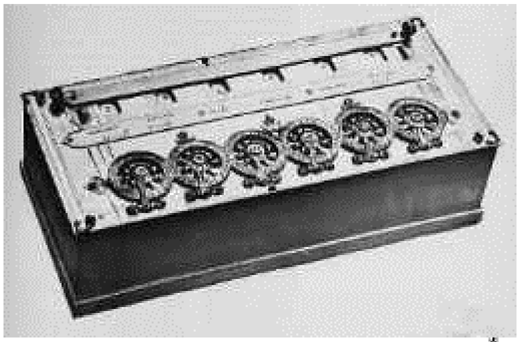


ব্লাইজ পাস্কাল

১৬৪২-এ, ফরাসি দার্শনিক কাম গণিতবিদ কাম পদার্থবিদ ব্লাইজ পাস্কাল (১৬২৩-১৬৬২) বানান একটা গণক যন্ত্র। ফ্রান্সের ক্লেরমঁ-ফেরান-এর এই বহুমুখী প্রতিভা পাস্কাল তার মাত্র ১৬ বছর বয়েসে জ্যামিতির একটা উপপাদ্য খাড়া করেন, যেটা এখনো পাস্কালস থিয়োরেম বলে পরিচিত। গণিতের তথা রাশিশাস্ত্রের তথা পদার্থবিদ্যার প্রোবাবিলিটি থিয়োরি বা সম্ভাবনা তত্ত্ব নিয়ে বড় মাপের কাজ করেন পাস্কাল। এমনকি তার কাজ ছিল আবহাওয়া বিদ্যা বা তরল পদার্থের ভৌত ধর্ম নিয়েও, পাস্কালস ল, পাত্রে আবদ্ধ একটা তরল তার সব দিকে সমান চাপ দেয়। সেই সময়টা মাথায় রাখুন, ফরাসি বৈজ্ঞানিক লাভোয়েজিয়ে যেমন গণিত করতে করতে ক্লান্ত হয়ে গেলে চলে যেতেন ইতিহাসের লাইব্রেরিতে, সেখানে ক্লান্ত হলে ভূগোলে, ইত্যাদি এবং ইত্যাদি।

পাস্কালের এই গণক যন্ত্রে বিয়োগ করা যেত। শ্যামনগরের বরদাখুড়ো যেমন আফিম খেয়ে ঝিমোতে ঝিমোতেই লেজারবুকে পেন রেখে বিড়বিড় করতেন, সাঁইত্রিশের সাত নাবে তিনে কত্তি তিন, সায়েব এসে কাঁধে চিমটি কেটে হেসে বলত, হ্যাভ এ কাপ অভ টী বাবু (কার্টসি পরশুরাম), সেইরকম পাস্কালের এই মেশিনও একটা স্তম্ভে বিয়োগ করে, আমরা যেমন পরের স্তম্ভের নিচের সারিতে এক যোগ করে নিই, পরের স্তম্ভে হাতে থাকা 'কত্তি তিন'-টা নিয়ে যেতে পারত।

এবং দেখুন, আবার সেই ট্যাক্স, ব্যাবিলন সিন্ধু থেকে পাস্কাল। পাস্কালের বাবা ছিলেন রুয়েনের ট্যাক্স কমিশনার। কাজের চাপে নাজেহাল। তরুণ পাস্কাল আসলে এই যন্ত্র বানিয়ে তার পিতৃঋণ শোধ করার চেষ্টা করেছিলেন। পাস্কাল কাজে হাত দেন ১৬৪২-এ, যখন তার বয়েস উনিশ। তিন বছরের মধ্যে তিনি তার মেশিন, পাস্কাল্যাঁ, 'নারী পাস্কাল', প্রথম কেজো মডেল নামিয়ে দেন। পাস্কাল্যাঁর ক্ষমতা ছিল অত্যন্ত সীমাবদ্ধ। যোগ এবং বিয়োগ। গুণ এবং ভাগ পাস্কাল্যাঁর আওতার বাইরে ছিল। পাস্কাল এর পেটেন্টও করেন। কিন্তু তৈরি হয়েছিল মাত্র হাতে গোনা কয়েকটা। কারণ, খরচ ছিল কম না, আর কাজের বেলায় খুব একটা ভরসা করা যেত না, প্রায়ই আটকে যেত।

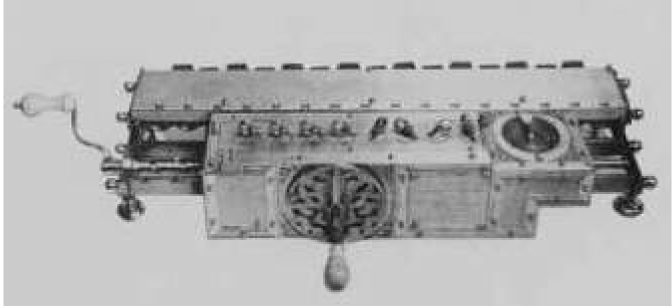


পাস্কালের পাস্কাল্যাঁ

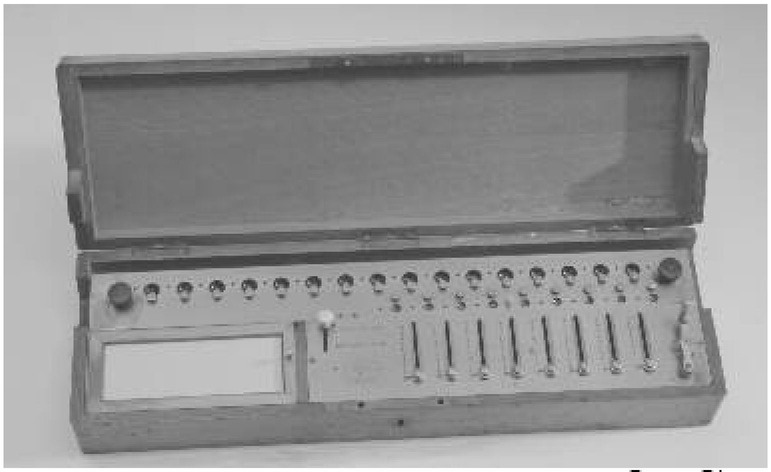
শিকার্ডের যন্তরটার কথা জানার আগে অব্দি পাস্কালকেই ধরা হত প্রথম ক্যালকুলেটর বানিয়ে। কিন্তু তাই বলে পাস্কালের কৃতিত্ব কোথাও ছোট হয় না। এমন হতেই পারে যে পাস্কাল শিকার্ডের ওই ক্যালকুলেটিং ক্লক মেশিনের কথা জানতেনই না। এছাড়া, আমাদের দিন শূন্যর যোগ বিয়োগ গুণ ভাগ-এর যুক্তিমালাটা মনে করুন। যোগ মানে দুটো জিনিষকে একই জায়গায় রাখা। গুণ মানে বারবার যোগ। ভাগ মানে গুণের উল্টো — ক দিয়ে খ কে ভাগ করে

ভাগফল গ খুঁজছি মানে, আসলে এমন একটা সংখ্যা গ খুঁজছি যাকে ক দিয়ে গুণ করলে খ হয়। কিন্তু বিয়োগ এর সম্পূর্ণ বিপরীত যাত্রা। যুক্তি কাঠামো মোতাবেক সম্পূর্ণ উল্টো একটা গতি। নাম্বার সিস্টেম গড়ে ওঠার ইতিহাসেও এটা একটা বড় পদক্ষেপ। এই মুহূর্তে মনে হচ্ছে, আগে একটু নাম্বার সিস্টেম বলে এলে ভালো হত। তখন ইচ্ছে করেই বাদ দিয়েছি। যাকগে।

ক্যালকুলাস আবিষ্কর্তা কাম বড়মাপের দার্শনিক গটফ্রিড উইলহেল্ম লিবনিতজ (১৬৪৬-১৭১৬) আবার পাস্কালের ওই পাস্কাল্যাঁ মেশিনের উপর একটা বিশেষ স্টেপড গিয়ার লাগানোর প্ল্যান করেন, যাতে সেই মেশিনে গুণ-ও করা যায়। এই করতে গিয়ে সেটা হয়ে ওঠে সম্পূর্ণ নতুন একটা যন্ত্র। লিবনিতজ এই যন্ত্রের নাম দেন স্টেপড রিকনার। এই স্টেপড রিকনার মেশিনে লিবনিতজ ব্যবহার করেন স্টেপড গিয়ার বা লিবনিতজ হুইল, যা পরবর্তী অনেক মেশিনেই ব্যবহৃত হয়েছে। লিবনিতজ-এর এই স্টেপড রিকনার ছিল পাস্কালের মেশিনের চেয়ে বহুগুণ জটিল। আর, আগেই বললাম, পাটীগণিতের মূল চারটে ক্রিয়া, যোগ বিয়োগ গুণ ভাগ — সব কটাই করতে পারত এই স্টেপড রেকনার। স্টেপড রেকনারের মেশিন ডিজাইনের মধ্যে একটা গঠনগত ত্রুটি ছিল যা ধরা পড়েছিল বহু বছর পরে।


লিবনিতজ-এর স্টেপড রেকনার

প্রথম সত্যিকারের কাজেবল ক্যালকুলেটর বানিয়েছিলেন চার্লস জাভিয়ার থমাস দ কোলমার (১৭৮৫-১৮৭০) ১৮২০ সালে। যার নাম দেন অ্যারিথমোমিটার। লিবনিতজ-এর ওই স্টেপড গিয়ার পদ্ধতিকেই কাজে লাগান কোলমার। পাটীগণিতের চারটে কাজই করতে পারত। গিয়ারের চাকাটা ঘুরত সামনের দিকে। এবং সেই অবস্থায় যোগ আর গুণ করতে পারত অ্যারিথমোমিটার। গিয়ারটা উল্টে দিতে হত বিয়োগ বা ভাগ করতে হলে। এই অ্যারিথমোমিটার যন্ত্রের পেটেন্ট করা হয় ১৮২০ সালের নভেম্বর মাসে। এর থেকে একটু একটু বদলে বদলে নতুন নানা মডেলের বিক্রির তথ্য পাওয়া যায় ১৯০০ সালের পরেও। এবং এর কয়েকটি ১৯৪০ সাল অব্দিও কাজ করত, তার প্রমাণ পাওয়া যায়। এই জাভিয়ার থমাস দ কোলমার সম্পর্কে আমার আর একটু কিছু জানার কৌতূহল রইল। আমার কাছে যা এনসাইক্লোপিডিয়া আছে তাতে পাইনি। নেটে আর খোঁজা সম্ভব না, এই পাঠমালার জন্যে আমার অর্থনৈতিক ধ্বস নেমে যাচ্ছে, আর বোধহয় উইদাউট পে হতে হবে এই ঠ্যাঙের ছুটিতে। আপনারা যদি কেউ কিছু পান, পাঠাবেন প্লিজ। একটা লিংক পেয়েছিলাম, গিয়ে দেখতে পারেন — http://vmoc.museophile.com।



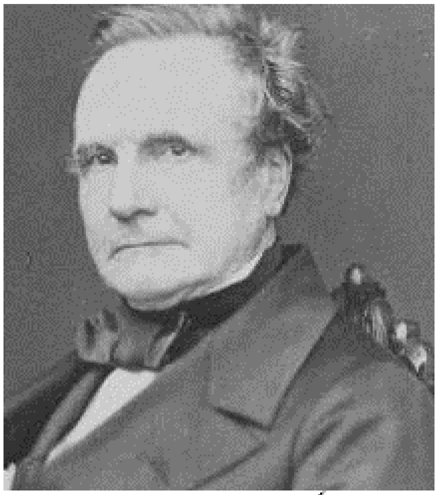
অ্যারিথমোমিটার

এইখানে আমাদের এই ১.৩ সেকশন শেষ হচ্ছে। এর পরেই ১.৪ সেকশনে আমরা যাব ব্যাবেজ-এ, তার ডিফারেন্স ইঞ্জিন এবং অ্যানালিটিকাল ইঞ্জিন-এ। এই সেকশনে আমরা ব্যাবেজের পূর্বসূরীদের দীর্ঘ তালিকা থেকে কয়েকটাকে জানলাম। আর একটা কথা, এখানে একটা মেশিনের আলোচনা করব কিনা তা নিয়ে আমি নিজেও সংশয়ে ছিলাম, সেটার কথা উল্লেখ করে রাখা যাক। এখানে উল্লেখিত প্রথম মেশিনের নির্মাতা শিকার্ড জন্মেছিলেন ১৫৯২-এ। বয়সে তার থেকে মাত্র একশো চল্লিশ বছরের বড় আর একজন, যেকোনো বিষয়েই প্রারম্ভিকতম কথাগুলো তিনি আগেভাগেই বলে রেখেছেন এটা যেন সভ্যতা আর সংস্কৃতির ইতিহাসে স্বতঃসিদ্ধ হয়ে গেছে — লিওনার্দো দ ভিঞ্চি (১৪৫২-১৫১৯), যথারীতি, তার নোটবইয়ে একটা ড্রয়িং করে রেখেছিলেন ক্যালকুলেটিং মেশিনের। সেই ড্রয়িংটা আবিষ্কৃত হয় অনেক পরে, ১৯৬৭-তে, স্পেনে। কোডেক্স মাদ্রিদ নামে পরিচিত এই প্ল্যানটা থেকে পরে একটা মেশিন বানানো হয়। মূল ড্রয়িং, পরিকল্পনা এবং তৈরি মেশিনের ছবি — গোটাটা খুবই চিত্তাকর্ষক। কিন্তু, এক, এটা সেই অর্থে লিওনার্দোর তৈরি নয়, তৈরি করা এই শতাব্দীর দ্বিতীয়ার্ধে। আর দুই, এর গোটাটাই ভীষণ ভাবে কপিরাইটের আওতায়। কপিলেফট কেন দরকার বুঝতে পারছেন? আমাদের এই জিএলটি ইশকুল পাঠমালার টেক্সটা তৈরি হচ্ছে যে কেউকে যে কোনো ভাবে প্রকাশের অধিকার দিয়ে, শুধু লেখায় কোনো পরিবর্তন আনা চলবেনা, এবং লেখার গোড়াতেই, গোটা লেখাটার ইলেকট্রনিক সংস্করণ পাওয়ার উৎস উল্লেখ করতে হবে। আমাদের পরিকল্পনা আছে এটা লাগ বা ওই ধরনের কোনো ওয়েবসাইটে রাখার। সঙ্কর্ষণ আর অশেষ খুবই চেষ্টাচরিত্তির চালিয়ে যাচ্ছে এটাকে হার্ডকপিতে মানা ছাপা সংস্করণে প্রকাশ করার, যে এফএসএফ হোক আর অন্য কেউ হোক। দেখা যাক কী হয়। কপিরাইট থাকায় এখানে দিতে পারলাম না, কিন্তু ভিঞ্চির মেশিনের ভারি চমৎকার ছবি আছে নেটে। লিংকটা হল, http://www.webcom.com/calc/Calculting\ Machines.html, গিয়ে দেখুন।

১.৪।। চার্লস ব্যাবেজ এবং অঙ্কের ইঞ্জিন

হার্ডওয়ারের ইতিহাসে এর আগের সেকশনের উদাহরণগুলোর মত অনেক দেশে অনেক জায়গায় অনেক চেষ্টার মধ্যে থেকে চার্লস ব্যাবেজের (১৭৯২-১৮৭১) অ্যানালিটিকাল ইঞ্জিন আলাদা হয়ে দাঁড়ায় একা চার্লস ব্যাবেজের জন্যে নয়, ব্যাবেজ ছিলেন এর হার্ডওয়ার ইঞ্জিনিয়র, একজন সফটওয়ার ইঞ্জিনিয়রও ছিলেন, যার নাম আডা লাভলেস, তার কথায় আসছি আমরা, একটু পরে। আজকের দিনের দ্বিতীয় যে অংশটা, কম্পিউটার চিন্তনের ইতিহাস, সেখানেও আসবে এই ব্যাবেজ-লাভলেস। তৃতীয় সৌরগ্রহের মাটিতে প্রথম সত্যিকারের ডিজিটাল বা আঙ্কিক কম্পিউটার নামিয়েছিলেন ইংরেজ অঙ্কবিদ চার্লস ব্যাবেজ।

> প্রথম খসড়ায় এখানে একটা ভারি উদ্ভট ভুল ঘটেছিল, তথাগত খেয়াল করিয়ে দিয়েছে। খুবই বিদঘুটে ভুল। সালটা একই ছিল, শুধু নামটা হয়েছিল অগাস্ট ব্যাবেল। নিজেরই অবাক লাগছে, জার্মানির সোশাল ডেমোক্রাট পার্টির প্রতিষ্ঠাতা অগাস্ট বেবেল, বা তার লেখা, আমার রাজনৈতিক সময়গুলোয়, খুব একটা দূরবর্তী ছিলনা, কিন্তু সেটা এতদিন পর, ওই বানান পার্থক্য সমেত? প্রায় দৈনন্দিন বেখেয়ালি ভুল নিয়ে ফ্রয়েডের থিসিসের একটা টেস্টকেস।



চার্লস ব্যাবেজ

ব্যাবেজের পক্ষে সবচেয়ে বড় ভুলটা হয়েছিল ভুল শতাব্দীতে জন্মানো। জীবনের শেষ দিকে তিক্ত ক্লান্ত ব্যাবেজ প্রায়ই বলতেন, জীবনে তো একটা দিনও আনন্দে কাটেনি, অবশিষ্ট জীবনটুকুর গোটাটাই দিয়ে দিতে রাজি যদি পাঁচশো বছর বাদে ভবিষ্যতের পৃথিবীতে তিনটে দিনও বাঁচার সুযোগ পাই। বড়লোক ব্যাঙ্কারের ছেলে ছিলেন চার্লস ব্যাবেজ, পড়াশুনো করেছেন ডেভনে, পরে কেমব্রিজে। যেখানে, তার মাস্টারমশাইদের চেয়ে তিনি বেশি অ্যালজেব্রা জানতেন। ১৮১২ সাল নাগাদ, বিশ বছর বয়সে, ব্যাবেজের মাথায় আসে, তারা যে লগারিদম এবং ট্রিগোনোমেট্রির টেবিল দিয়ে অঙ্ক কষেন, তার গোটাটাকে একটা যন্ত্র দিয়ে করানো যায়। সেইখান থেকে শুরু তার অসম্পূর্ণ তাই অসফল আজীবন যাত্রার।

ব্যাবেজের কালে, খুব খুচরো দুধেভাতে যোগ বিয়োগ বাদ দিলে, আর সব হিশেবের কাজই করা হত লগটেবিল দিয়ে। নাবিকরা, জমি জরিপের লোকেরা, জ্যোতির্বিদরা — সবারই হাতে থাকত পেন্সিল আর লগটেবিল, আর ত্রিকোনমিতির সাইন কস ট্যান এসবের মানের জন্যে ট্রিগোনোমেট্রিক টেবিল। এই টেবিল ছাড়া এক পা এগোনোর উপায় থাকত না। কিন্তু মজার কথা, এই টেবিলগুলো হত ভুলের ধাঁধায় ভরা।

আসলে এই লগারিদম, ট্রিগোনোমেট্রি আর অ্যাস্ট্রনমির টেবিল তৈরি-করা তখন ইংলন্ডে প্রায় একটা কুটীরশিল্প। ঘরে ঘরে লোক বসে বসে পাতার পর পাতার পর পাতা হিশেব কষে যাচ্ছে, টেবিল বানাচ্ছে। আমাদের ছোটবেলায় লোকে যেমন করে সিএ পড়ত। কার কাছে একটা শুনেছিলাম, তার গোটা ছেলেবেলাটা কেটেছে সিএ-র দুঃস্বপ্ন নিয়ে, যতক্ষণ সে জেগে থাকত, অহর্নিশি দেখত, ঠিক সামনের বাড়ির জানলায় একটা মুন্ডু ঝুঁকে রয়েছে খাতায়, সিএ পড়ছে। এটা চলেছিল তার বিএ পাশ এবং বিয়ে করা অব্দি, কারণ, তারপর সেই আশু-সিএ-সম্পন্ন পরিবার অন্য পাড়ায় চলে গেছিল বাড়ি ভাড়া নিয়ে। ইংলন্ডে শিল্পবিপ্লবের এই গোটা সময়টা জুড়ে চার্চ বা গির্জে সমাজ জীবনে একটা বিরাট জায়গা নিয়ে বিরাজ করত, অর্থনীতির প্রতিটা স্তরে, প্রতিটা গ্রামে, প্রতিটা প্যারিশে। গাঁয়ের গির্জেতে গির্জেতে পাদ্রিদের সময় কাটানোর মানে ছিল অঙ্ক করা, অর্থাৎ, মূলত হিশেব কষা। নাপিয়ের নিজে সম্ভ্রান্ত অভিজাত ছিলেন, পাদ্রি ছিলেন না, কিন্তু তারও ব্যাপারটা ছিল এই একই।

ওদিকে ফরাসি বিপ্লবের পর ফরাসি সরকার একটা নতুন ধরনের ব্যবস্থা আনল। পুরোনো এককগুলো বাতিল করে দশমিক বা ডেসিমাল সিস্টেমে নিয়ে আসতে শুরু করল সমস্ত একককে। সিজিএস, সেন্টিমিটার গ্রাম সেকেন্ডের, সেকেন্ড মানে সময় ভাগটা বাদ দিলে অন্য দুটোই এই দশমিক ব্যবস্থা। তাপমাত্রা ফারেনহেইট থেকে সেন্টিগ্রেড, ওজন পাউন্ড থেকে কেজি, তরলের আয়তন গ্যালন থেকে লিটার, দূরত্ব মাইল থেকে কিলোমিটার। দশের ভাগে ভাগে। আমরা ছোটবেলায় মুখস্থ করতাম, মিলি সেন্টি ডেসি হুঁঃ ডেকা হেক্টো কিলো। হুঁঃ মানে একটা অবস্থান — এখানে গ্রাম লিটার মিটার সবই আসতে পারে। আর প্রত্যেকটা স্তর তার নিচের স্তরের দশগুণ। ডেসিমাল ব্যবস্থা।

ব্যাপারটা এতদূর এগিয়েছিল যে এইসময় ফ্রান্সে দশটা দাগের ডায়ালের ঘড়িও চালু হয়, আমরা যেমন বারোটা দাগের ডায়ালের ঘড়ি ব্যবহার করি। এই দশভুজ আননের ঘড়ি আজো কোনো কোনো মিউজিয়ামে আছে। কিন্তু

হিশেবের জগতে এতে ঘটলটা এই যে সমস্ত পুরোনো হিশেবের টেবিল নতুন করে কেচে গণ্ডূষ করতে হল, দশ-দিক থেকে এই দশ-মিক আক্রমণে সমস্ত পুরোনো হিশেব তো ঘেঁটে গেছে — সমস্ত পরিমাপ তো ডেসিমাল হয়ে গেছে, বদলে গেছে।

তার মানে একদম নতুন করে সব টেবিল বানাতে হবে। নতুন করে ওই তাল তাল হিশেব। এবং সেগুলো যদি যথেষ্ট সঠিক না-হয়, সমাজ-ইতিহাসের প্রতিটা স্তরেই কেরোসিন, বাজে মাপে বানানো বাজে ম্যাপের দৌলতে উপনিবেশ বানানোর দৌড়ে ইউরোপের অন্য জাতিগুলোর সঙ্গে হেরে যাওয়াটা তো ছেড়েই দিন। ফরাসি সরকারের কাছ থেকে এই নতুন টেবিল বানানোর দায়িত্ব পেয়েছিলেন ব্যারন দ প্রনি। প্রনি তার এই দৈত্যাকৃতি কাজকে তিনটে টুকরোয় ভেঙে নিয়েছিলেন প্রনি। এক, কিছু নামজাদা গণিতবাজ ডিসিশন নেবে ঠিক কোন কোন ফরমুলা দিয়ে কী কী করতে হবে এবং কী ভাবে। এরা হল প্রনিদলের মুণ্ড। দুই, একটা বিরাট সংখ্যক গণিতজ্ঞ এই তথ্যের মূল কাঠামোটা খাড়া করবে — এরা হল ধড়। আর তিন, আশিজন তুখোড় হিশেবী, যাদের হিউম্যান কম্পিউটার বলে, দিনরাত কষতে থাকবে সংখ্যাগুলো। প্রনির টিমের এরা হল পা, হাঁটিয়ে নিয়ে যাবে প্রনির গোটা প্রোজেক্টটাকে।

আজকের দিনে দাঁড়িয়ে আমাদের ফিল করাটাও বিতিকিচ্ছিরি রকমের শক্ত, হিশেবপরাক্রমী এই কাকেশ্বর কুচকুচদের লাইফ কতটা হেল হয়ে যেত — এই 'সামনে রইল টেবিল আর হাতে রইল পেন্সিল' প্রক্রিয়ায়। "ভাই আমি তো আমার ছেলে হওয়ার দিনে ভেবেছিলাম, নাতির অন্নপ্রাশনের মধ্যে আমার ট্রিগোনোমেট্রির টেবিলটা শেষ করতে পারব, এখন দেখছি, নাতির বিয়ের আগে হবে না, আসলে বয়েস হচ্ছে তো, স্পিড কমে যাচ্ছে রে ভাই"।

লগারিদম আর ট্রিগোনোমেট্রি আর অ্যাস্ট্রোনমির টেবিল — এদের কষে তুলতে গিয়ে এত এত মানুষের এত এত শ্রম চলে যাচ্ছিল যে শ্রম বাঁচানোর জন্যে তারা অনেক রকম কায়দাকানুন আয়ত্ত করেছিল। এর মধ্যে একটা কায়দা ছিল, কয়েকটা স্টেপ অব্দি মান হিশেব করে নিয়ে, তারপর, ওই মানগুলোর সঙ্গে মিলিয়ে অন্য মানগুলো ভরে দেওয়া। এই ভরার জন্যে একটা চালু উপায় ছিল রেখা-লাগানো বা কার্ভ-ফিটিং। তারপর, ওই কার্ভ আসতে পারে যে ফাংশন থেকে, সেই ফাংশনটা অঙ্ক কষে বার করে ফেলা। এই কাজে খুব বেশি ব্যবহার হত পলিনোমিয়াল ফাংশন। একটু বাদে ডিফারেন্স ইঞ্জিনের কাজের আলোচনায় পলিনোমিয়াল ফাংশনে আসছি আমরা।

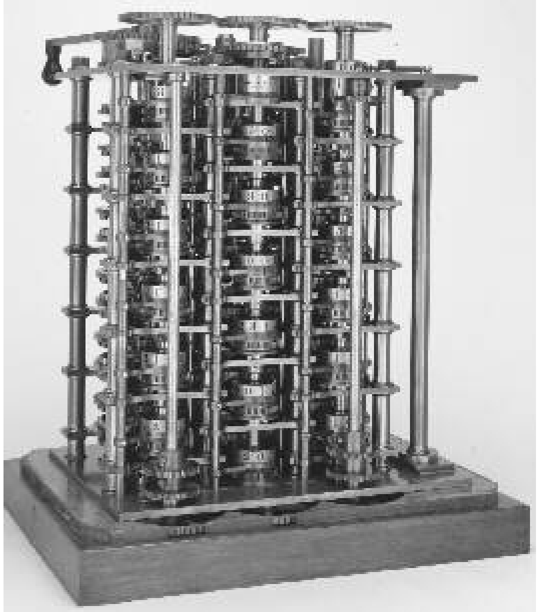
এবার ওই কার্ভের উপর ফিট করা পলিনোমিয়ালটার পরপর ভ্যালু বার করে যাও। এবং, সাবধানের মার নেই, মাঝে মাঝে কিছু কিছু জায়গায় হিশেব করে দেখো, সেটা মিলছে কিনা, বা, কতটা মিলছে। যদি একান্তই না-মেলে, কুছ পরোয়া নেই। কোন একজন বিখ্যাত ডুবুরি বলেছিলেন না, একই নদীতে আমরা দুবার ডুব দিতে পারিনা। একটা পলিনোমিয়াল মিলল না, এ আর এমন কি, আর একটা পলিনোমিয়াল আর একবার কষে নাও। কষে দেখো, নতুন কোন পলিনোমিয়াল কার্ভটার কাছাকাছি আসছে। নতুন কষা পলিনোমিয়ালটা এবার কার্ভের নতুন ফিটিং। এতে সংখ্যার দেবতার খেরোর খাতায় একটু পাপ হচ্ছে ঠিকই, কিন্তু পাপ আর পুণ্যের পার্থক্য তো দশমিকের পরের কয়েক ঘর অব্দি। সেই অব্দি মিলে গেলে, পাপ আর পুণ্য দুই-ই পরম ব্রহ্মে লীন হয়ে যায় রে ভাই। কাইন্ডলি দেখো, দশমিকের পরে ওইটুকু অব্দি যেন মেলে। দশমিকের পরে মোটামুটি কয়েক ঘর অব্দি একটা গাণিতিক মান মোটামুটি ঠিকঠাক এসে গেলে, তারপর আমাদের আর কাজের অসুবিধে হওয়ার কথা নয়। রাশিশাস্ত্রে বা স্ট্যাটিস্টিক্সে নিউমেরিকাল অ্যানালিসিস বা সংখ্যা ভিত্তিক কাজের ধারাটাও অনেকটাই এই। এবার, একবার পলিনোমিয়াল ফাংশনটা পেয়ে গেলাম মানে, তারপরে সেটাকে ডিফারেন্স মেথড বা পার্থক্য পদ্ধতি দিয়ে পরপর কষে যাওয়া। ডিফারেন্স মেথডের কথায় আমরা এখুনি আসছি।

ও, একটা কথা, এই পলিনোমিয়াল ফাংশন দিয়ে যে কোনো লগারিদমিক বা ট্রিগোনোমেট্রিক ফাংশনকে যথেষ্ট সঠিকভাবে অ্যাপ্রক্সিমেট করা যায়, বা মেলানো যায় ঠিকই, কিন্তু হঠাৎ আমরা ট্রিগোনোমেট্রিক ফাংশন আর লগারিদমিক ফাংশনের কার্ভকে ট্রিগোনোমেট্রিক আর লগারিদমিক ফাংশন দিয়ে না-কষে পলিনোমিয়াল ফাংশন দিয়ে কষছি কেন? কারণ, পলিনোমিয়াল মানে একটা বীজগাণিতিক সমীকরণ, যাকে সমাধান করা যায়, কষে ফেলা যায়, তুলনামূলক ভাবে অনেক সহজেই। যা ট্রিগোনোমেট্রিক বা এক্সপোনেনশিয়াল বা লগারিদমিক সমীকরণের বেলায় একেবারেই সত্যি নয়। সেগুলো প্রায়ই হয় রীতিমত হাতা — দেখলেই জ্বর এসে যায়, কষা তো দূরের কথা।

প্রনির টিমের এই হিউম্যান কম্পিউটারদের কাজ ছিল মূলত মূলত বিদ্যুৎবেগে ঘচাঘচ করে গাবদা গাবদা যোগ আর বিয়োগ কষে যাওয়া। এরা এসেছিল বিভিন্ন ধরনের জীবিকা থেকে। কিছুদিন আগেই, ফরাসি বিপ্লবের আগে আগেই, হয়ত কোনো নবাবের, মানে ফরাসিদের তো নবাব হয়না, কোনো কঁত ভিসকঁত মারকুইর চুল ছাঁটত ফ্যাশানেবল করে এমন লোকও ছিল প্রনির বাহিনীতে। এদের শুধু যোগ আর বিয়োগ দ্রুতবেগে কষতে পারলেই হত, কারণ এরা কাজ করত মেথড অফ ডিফারেন্স বা পার্থক্যের পদ্ধতি দিয়ে।

এই হিউম্যান কম্পিউটারদের হয়ে ডিফারেন্স পদ্ধতির এই গোটা কাজটা মেশিন দিয়ে করে ফেলার পরিকল্পনা করেছিলেন ব্যাবেজ, সেই জন্যেই পরিকল্পিত মেশিনের নাম দিয়েছিলেন 'ডিফারেন্স ইঞ্জিন'। আমরা এখানে ১৮৩২-এর ডিফারেন্স ইঞ্জিনের একটা অংশের ছবি দিয়েছি, তাও এটা প্রথম নির্মাণের, ব্যাবেজ এটা তৈরি করে গেছেন, বারবার, টুকরোয় টুকরোয়। কতটুকু তিনি করতে পারছেন, তার গোটাটার কী সাফল্য — কী প্রভাব ফেলবে সেটা কম্পিউটারের তথা বিজ্ঞানের সামগ্রিক ইতিহাসে সেটাও জেনে বুঝে উঠতে পারেননি ব্যাবেজ। অনেক উপর থেকে বিমান দিয়ে ছাড়া যেমন যেমন চীনের পাঁচিল দেখা বা বোঝা যায়না। বা, পেরুতে ইংকাদের আঁকা ওই কয়েক কিলোমিটারের ধনুর্বীর যেমন। আজকে যে আমরা ব্যাবেজের কাজের লাভলেসের কাজের সঙ্গে আত্মীয়তা বোধ করব যে সেটা ব্যাবেজ নিজেও জানতেন না। জানতেন না যে ভবিষ্যতের এই পৃথিবীতে তিনদিনের চেয়ে অনেক বেশিদিন আরামসে বেঁচে থাকবেন। ব্যাবেজ বিপুলভাবে রয়েছেন, টেক্সটে টেক্সটে মেশিনে মেশিনে ছড়িয়ে, শ দুয়েক বছরের পরের এই পৃথিবীতে।

ডিফারেন্স ইঞ্জিন



ডিফারেন্স ইঞ্জিনের কাজের এলাকাটা একটু বোঝার চেষ্টা করা যাক। ধরুন আমরা একটা পলিনোমিয়ালের মান জানতে চাইছি। পলিনোমিয়াল বলতে আমরা বুঝি একাধিক খণ্ডের সমষ্টি একটা গাণিতিক চিহ্নমালাকে, যেখানে প্রতিটা খণ্ডের মধ্যে থাকে একটা চলরাশি, ধরুন ক, তার কোনো একটা পাওয়ারে বা সূচকে, ধরুন ২ বা ৩ বা ৪ ইত্যাদি। আর এই প্রত্যেকটা ক বা ক-এর পাওয়ারের সঙ্গে থাকে একটা গুণক মানে স্থির সংখ্যা। যেমন ধরুন এই পলিনোমিয়ালটা —

$$ক + ২ক^{২} + ৩ক^{৩}$$

এর মানটা আমরা জানতে চাই, ক-এর অনেক অনেক মানের জন্যে, ধরুন ক = ১, ক = ২, ক = ৩, . . ., ইত্যাদি। এটাকে আমরা পুরোটা গুণ করে করে বার করতেই পারি। কিন্তু সেটা অনেক পরিশ্রমসাধ্য। ধরুন ক = ৭ এই অবস্থায় আমি যদি পলিনোমিয়ালটার মান পেতে চাই, আমাকে এই গোটাটা হিশেব করতে হবে —

$$৭ + (২ \times ৭ \times ৭) + (৩ \times ৭ \times ৭ \times ৭)$$

প্রত্যেকবার, ক-এর প্রতিটি আলাদা আলাদা মানের জন্যে এই গোটা হিশেবটা কষাই এর একমাত্র ব্যথা নয়। এর সঙ্গে আরো বড় ঝামেলাটা হতে পারে এই যে, ক-এর যে মানের জন্যে হিশেবটা কষতে চাইছি, সেটা একটা বিতিকিচ্ছিরি রকমের বড় সাইজের সংখ্যা, কয়েক টনের। ধরুন, সাত না হয়ে সাত লক্ষ সাতাত্তর হাজার সাতশো সাতাত্তর। মেথড অফ ডিফারেন্স বা পার্থক্যের পদ্ধতি দিয়ে এই গোটাটাই করে ফেলা যায় নিছক যোগ আর বিয়োগ করে। নিচের তালিকাটা দেখুন, এখানে আমরা ক-এর মান ১ থেকে ৫ অব্দি আমাদের ওই পলিনোমিয়ালটার মান বার করেছি — ক + ২ ক^{২} + ৩ ক^{৩}

ক	ক + ২ ক^{২} + ৩ ক^{৩}	১ম পার্থক্য	২য় পার্থক্য	৩য় পার্থক্য
১	৬	—	—	—
২	৩৪	২৮	—	—
৩	১০২	৬৮	৪০	—
৪	২২৮	১২৬	৫৮	১৮
৫	৪৩০	২০২	৭৬	১৮

এখানে প্রথম পার্থক্যটা হল পলিনোমিয়াল ফাংশনটার পরপর দুটো মানের ভিতরকার অন্তর বা দূরত্বটা। যেমন ২৮ হল ৬ আর ৩৪-এর ভিতরকার দূরত্ব, ৬৮ হল ১০২ আর ৩৪-এর ভিতরকার দূরত্ব, ১২৬ হল ২২৮ আর ১০২-এর দূরত্ব, ইত্যাদি, পরপর করে যান। দেখুন, আমরা এখানে দূরত্ব হিশেবে বলছি, তার মানে ফাংশনটা যদি ক্রমশ কমতে থাকে তাহলে পরেরটা আগেরটার চেয়ে ছোট হবে, তার মানে এখানে যেমন ৩৪ থেকে ৬ বাদ দিয়ে একটা ধনাত্মক সংখ্যা পেয়েছি, সেরকম আর পাব না, তখন বিয়োগ করে পাব ঋণাত্মক সংখ্যা। ধনাত্মক না ঋণাত্মক তাতে কিছু এসে যায় না, আমরা শুধু তার মানটা নেব, গণিতে যাকে বলি মডুলাস বা অ্যাবসলিউট বা চূড়ান্ত মান। দ্বিতীয় পার্থক্য মানে প্রথম পার্থক্যগুলোর ভিতরকার দূরত্ব। যেমন ৬৮ আর ২৮-এর দূরত্ব হল ৪০। ১২৬ আর ৬৮-র দূরত্ব হল ৫৮। ২০২ আর ১২৬-এর দূরত্ব ৭৬। আবার তৃতীয় পার্থক্য হল দ্বিতীয় পার্থক্যদের পার্থক্য, ৫৮ আর ৪০-র পার্থক্য হল ১৮, আবার ৭৬ আর ৫৮-র পার্থক্য ১৮। এখানে এসে প্রথম এই মিলটা তৈরি হল, এর আগে অব্দি যে কোনো একটা পার্থক্যের সিরিজে পরপর মানগুলো হচ্ছিল আলাদা, এখানে এসে পরপর মানগুলো সমান পেলাম। কেন হল এরকম, ভেবে দেখুন তো, কেন তিন নম্বর পার্থক্যে এসেই পেলাম আমরা সবগুলোকে সমান? এখানে ক-এর সর্বোচ্চ সূচক তিন না হয়ে চার হলে কত নম্বর পার্থক্যে গিয়ে সিরিজটা পরপর একই সংখ্যার পেতাম বলে মনে হয়?

এবার তালিকায় দেখুন ক = ৪ আর ক = ৫ এই দুটোর জন্যেই তৃতীয় পার্থক্য এসেছে ১৮। দম থাকলে করে যান, দেখবেন এরপরেকার ক-এর যাবতীয় মানের জন্যেই এই পার্থক্যটা আসবে এই ১৮। এই চূড়ান্ত পার্থক্যটার মান এই দুঃসহ ১৮ না হয়ে বিনীত বিয়াল্লিশ বা প্রমিজিং পাঁচ হলনা কেন সেই দোষ আমাদের না দিয়ে দিন ওই ব্যাটা পলিনোমিয়ালকে। অন্য আর একটা পলিনোমিয়াল ফাংশনে অন্য একটা মান পাব এই তৃতীয় তথা অপ্রশ্নেয় পার্থক্যের। এই অনড় পার্থক্যটার মান পেয়ে গেলাম মানেই এক অর্থে কেল্লা ফতে। এবার এটা যে কোনো দ্বিতীয় পার্থক্যে যোগ করে দিলেই পাব তার পরের দ্বিতীয় পার্থক্যটা। এবার সেই দ্বিতীয় পার্থক্যটা সেই সারির প্রথম পার্থক্যের সঙ্গে যোগ করে দিলেই পাব পরবর্তী প্রথম পার্থক্য। এই ভাবে পলিনোমিয়ালটার পরবর্তী মানগুলো, ক-এর পরের পরের মানের জন্যে, পেয়ে যেতে পারব।

৩য় পার্থক্য	২য় পার্থক্য	১ম পার্থক্য	ক + ২ ক^{২} + ৩ ক^{৩}	ক
১৮	৭৬	২০২	৪৩০	৫
—	+ ১৮	+ ৯৪	+ ২৯৬	—
—	= ৯৪	= ২৯৬	= ৭২৬	৬

তাই, শেষ দ্বিতীয় পার্থক্য ৭৬ এর সঙ্গে ১৮ যোগ করে পাচ্ছি ৯৪, এর পরের দ্বিতীয় পার্থক্য। ৯৪ এর সঙ্গে শেষ প্রথম পার্থক্য ২০২ যোগ করে পাচ্ছি ২৯৬, পরের প্রথম পার্থক্য। এবার এই ২৯৬ যদি ক = ৫ এর জন্যে পলিনোমিয়ালের যে মান, মানে ৪৩০, তার সঙ্গে যোগ করে দিই, আমরা পাই ক = ৬ এর জন্যে পলিনোমিয়ালটার মান। এই ভাবে পর পর ক = ৭, ৮, ৯ . . . আমরা পর পর পেয়ে যেতে পারি পলিনোমিয়ালটার মান। একেই বলে পার্থক্যের পদ্ধতি বা মেথড অফ ডিফারেন্স। এখন দেখুন আর পলিনোমিয়ালটার মান পেতে আমাদের ওই অমানুষিক হিশেব আর করতে হচ্ছেনা। গুণের বৈগুণ্য থেকে ক্লিয়ারকাট ফগুবেনে যোগব্যায়াম, তাতেই ওইসব দুর্ধর্ষ ফিগার।

এই সেই ডিফারেন্স মেথড যা দিয়ে কাজ করে ডিফারেন্স ইঞ্জিন। ধরুন ব্যাবেজের ওই ইঞ্জিনের একজন অপারেটর রয়েছে, সে চাকাগুলোকে সেট করে দিল ১৮, ৭৬, ২০২, এবং ৪৩০-এ। এবার প্রয়োজনীয় হাতল ঘুরিয়ে দিল। ইঞ্জিন এবার পর পর সারিবদ্ধ যোগগুলো করে চলবে নিজে নিজেই। এবং একটু বাদেই দিয়ে দেবে চূড়ান্ত সমাধান — ক = ৬ এর জন্যে পলিনোমিয়ালের মান ৭২৬। ভেবে দেখুন, একটু আগে অব্দি একটা চাকায় গিয়ারে চলা ক্যালকুলেটর ব্যাপারটাকে যতটা অসম্ভব মনে হচ্ছিল, এখন আর লাগছে?

ব্যাবেজের কাছে মূল আগ্রহ ছিল ভুল দূর করা। সচরাচর এই হিউম্যান কম্পিউটারদের করা টেবিলগুলোয় মূলত তিন ধরনের ভুল থাকত।

এক, হিশেবের ভুল, যা ইঞ্জিনের কখনো হবেনা।

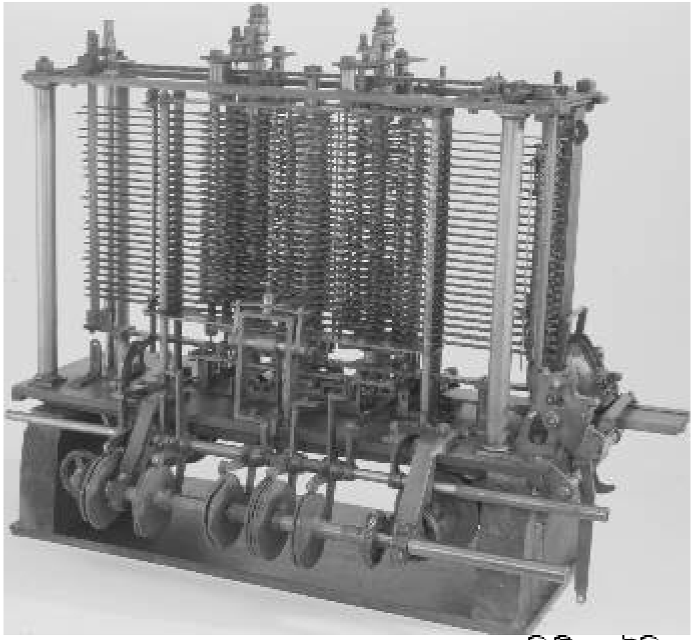
দুই, হিশেব করে পাওয়া ফলাফল টুকতে ভুল করা, এটাও ইঞ্জিনের বেলায় হওয়ার নয়।

তিন, এই ভুলটায় ইঞ্জিনের কিছু করার নেই, তৈরি হয়ে যাওয়া তালিকা ছাপার সময় অক্ষর বসাতে আর প্রুফ দেখতে ভুল হওয়া। এটারও একটা সমাধান ভেবেছিলেন ব্যাবেজ। সরাসরি যদি ইঞ্জিন একটা পাতার পুরো ছকটা তৈরি করে দেয়, তার থেকে তার ব্লক করে ছাপিয়ে ফেলা যাবে। এক্ষেত্রে তিন নম্বর ভুল হওয়ার সম্ভাবনাটাও থাকবে না।

ব্যাবেজ প্রথমে ডিফারেন্স ইঞ্জিনের একটা ছোট মাপের মডেল বানিয়েছিলেন, যাতে এর কাজ করার ব্যাকরণটা বোঝা যায়। ১৮২৩-এ ব্যাবেজ ব্রিটিশ সরকারের কাছ থেকে ১৫০০ পাউন্ডের একটা গ্রান্ট পান, তখন ব্যাবেজের প্ল্যান ছিল এমন একটা মেশিন বানানো যা ছ-নম্বর মানে সিক্সথ ডিফারেন্স অব্দি হিশেব করতে পারবে, এবং কাজ করতে পারবে কুড়িটা অব্দি অঙ্ক নিয়ে। তার সময়ের টেকনোলজির পক্ষে এবং তার খরচের পরিমাণের পক্ষে এটা ছিল বড্ড বেশি একটা চাহিদা।

সময় যাওয়ার সঙ্গে সঙ্গে ব্যাবেজ সরকারের কাছ থেকে আরো আরো গ্রান্ট পেয়েছেন, এবং নিজের সম্পদেরও একটা বড় অংশ চলে গেছে এই ইঞ্জিনের বয়লারে। ১৮৩৩-এ ব্যাবেজের সঙ্গে তার যন্ত্রনির্মাতা জোসেফ ক্লিমেন্টের গন্ডগোল বাধে এবং ক্লিমেন্ট কাজ ছেড়ে চলে যান। রগচটা ক্রোধের জন্যে আজীবন স্বনামধন্য ছিলেন ব্যাবেজ। এই ডিফারেন্স ইঞ্জিনের কাজ চলতে চলতেই ব্যাবেজের মাথায় আসে সেই উজ্জ্বল আইডিয়া, আজকের ডিজিটাল কম্পিউটারের যা পূর্বসূরী — অ্যানালিটিকাল ইঞ্জিন। ব্যাবেজ তার অবশিষ্ট জীবন যৌবন ধন মান ব্যয় করে ফেলেছিলেন এই অ্যানালিটিকাল ইঞ্জিন বানানোর অধরা স্বপ্নে।

একবার মাথায় অ্যানালিটিকাল ইঞ্জিনের স্বপ্ন আসা মাত্রই এই নতুন ইঞ্জিনে আক্রান্ত ব্যাবেজ এবার প্রায় বিস্মৃত হলেন দুয়োরাণী ডিফারেন্স ইঞ্জিনকে। টেবিল মেবিল বানানোর ওসব তুশ্চু কথা কার মনে থাকে — অমরত্বের দিকচক্রবালে তখন ডাক পড়েছে, টেবিলে ছাই হবে কী। আর অ্যানালিটিকাল হল সেই দশদিকে আমন্ত্রণ পাঠানো চোরাবালির মত, ঘোড়সওয়ার যার অঙ্গে অঙ্গীকার দেয়নি কোনোদিন। ব্যাবেজের জীবৎকালে ব্যাবেজ একে কার্যরত দেখে যেতে পারেননি, আর আডা তো নয়ই। এই অ্যানালিটিকাল ইঞ্জিন কিন্তু কখনোই ডিফারেন্স ইঞ্জিনের পরের স্টেপ বা পরের মডেল ছিলনা। ছিল একদম ভিন্ন সম্পূর্ণ অন্য মাপের একটা ধারণা। ডিফারেন্স ইঞ্জিন ছিল মূলত একটা যোগ করার কল, যে কিছু বিশেষ দক্ষ এবং দ্রুত প্রকারে তার এই যোগ-বিয়োগের কাজ করতে পারে। খুবই সৃষ্টিশীল, কিন্তু বৈপ্লবিক নয় কখনোই। অ্যানালিটিকাল একদম আলাদা রাস্তা। এটা ঠিক যে, ডিফারেন্সে কাজ করতে করতেই ব্যাবেজ অ্যানালিটিকালে পৌঁছেছিলেন, কিন্তু কখনোই শুধু একটা যৌক্তিক যাত্রায় একটা থেকে অন্যটায় পৌঁছনো যায়না, চিন্তার স্তরে একটা সম্পূর্ণ উত্তরণ প্রয়োজন হয়।



অ্যানালিটিকাল ইঞ্জিন

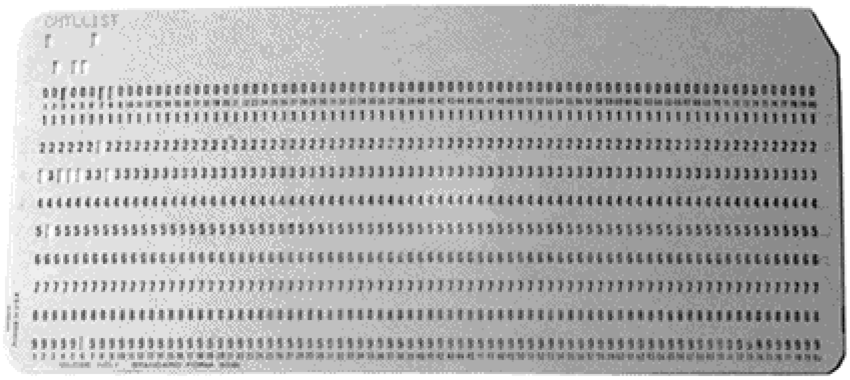
তার ডিফারেন্স ইঞ্জিনের খিঁচগুলো সবচেয়ে ভালো করে জানতেন ব্যাবেজ নিজেই। ধরুন, এই ইঞ্জিন একটা কার্ভ বেয়ে, পলিনোমিয়ালের একটা নির্দিষ্ট গঠন মোতাবেক ভ্যালু বার করেই চলেছে, করেই চলেছে, কার্ভের সঙ্গে মিলিয়ে মিলিয়ে। এবার, চলতে চলতে, কোনও এক সময় এই পলিনোমিয়াল মূল রেখা থেকে দূরে চলে যেতে শুরু করল। কিন্তু এবার কী করে বাধ্য করা যাবে ইঞ্জিনকে তার মূল যাত্রাপথে মানে ওই কার্ভে ফেরত যেতে? কী করে ইঞ্জিনকে বাধ্য করা যাবে নিজের কাজের ধারাটাকে বদলে নিতে? বদলাতে তো হবেই, কারণ, কার্ভের ওই নতুন রকমের অংশকে ব্যাখ্যা করতে লাগবে পলিনোমিয়ালের একটা নতুন গঠন। কিন্তু তার মানে মেশিনকে থামতে হবে, মেশিনের চাকাগুলো ফের নতুন করে অ্যাডজাস্ট করে নিতে হবে, নতুন পলিনোমিয়ালের মান অনুযায়ী। তাহলে আবার ফের নতুন-করে সেট-করা মেশিনের নতুন পলিনোমিয়াল সঠিক কার্ভ বেয়েই চলবে, যতক্ষণ না আবার একটা নতুনতর পার্থক্য গজিয়ে ওঠে কার্ভ আর পলিনোমিয়ালের ভিতর। তার মানে আবার একবার নতুন করে অ্যাডজাস্ট করো। এই ভাবে চলতেই থাকবে। এই নিরন্তর পুর্নমূল্যায়নের ঘাপলাটা চটিয়ে দিচ্ছিল ব্যাবেজকে। এক, কোথাও একটা নিজের মেশিনকে, বোধহয় তথা নিজেকেই, নড়বড়ে লাগছিল তার। তার মেশিন যেন পুরোপুরি নিজের কাজ করে উঠতে পারছে না, বারবার সাহায্যের দরকার পড়ছে। আর দুই, এই ফিরে চাওয়া কেন ফিরে ফিরে চাওয়া থেকে গজিয়ে উঠতে পারে নতুন অনেকতর অনাবশ্যক ত্রুটি। অ্যাডজাস্ট তো করবে মানুষ, তার মানে আবার একটা নতুন করে ভুলের উৎস, প্রাথমিক ভাবে যে ভুলের রাজত্ব থেকে সরে আসার উদ্দেশ্যেই এই ইঞ্জিনদের প্ল্যান।

১৮৩৪ থেকে ১৮৩৬ অব্দি ব্যাবেজ বারবার করে নতুন করে তার অ্যানালিটিকাল ইঞ্জিনের ডিজাইন বদলাতেই থাকলেন। আর এই প্রক্রিয়া চলতে চলতে, ব্যাবেজ, এবং ব্যাবেজের পাশাপাশি, বোধহয় আরো জোরদার রকমে, আডা বায়রন লাভলেস, ক্রমে যুক্তি বা লজিকের স্তরে আবিষ্কার করলেন আজকের একটা ডিজিটাল কম্পিউটারের মূল অংশ এবং কাজগুলোকে। ডিফারেন্স ইঞ্জিনের ঢাক ব্যাবেজ আজীবন পিটিয়েই গেছেন, কিন্তু অ্যানালিটিকালের ব্যাপারে তিনি প্রথম থেকেই মনসা চিন্তিতং কর্মং বচসা ন প্রকাশয়েত। হয়তো এর মধ্যে লোককে বলা এবং না-বুঝিয়ে উঠতে পারার তিক্ত হতাশাও কিছুটা থেকে থাকবে। তখন তার কাছে উদ্দেশ্য একটাই, চিন্তাকে যন্ত্রের জগতে অনুবাদ করা, গোটা চিন্তা না হোক, চিন্তন প্রক্রিয়ার এক একটা খণ্ডকে। গাণিতিক চিন্তাকে। ঠিক আজকের একটা প্রোগ্রাম যা করে। খণ্ডটার জটিলতার মাত্রা যাই হোক, কম বা বেশি।

অ্যানালিটিকালকে ব্যাবেজ কতকগুলো অংশের সমাহারে ভেবেছিলেন। একটা অংশের নাম 'মিল' বা কল, সেটা হল হিশেব করার কলকব্জা। একটা অংশ 'স্টোর', বা ভাঁড়ার। সংখ্যাখচিত চাকাগুলোকে বারবার অ্যাডজাস্ট করা হচ্ছে। প্রতিবার তাদের অ্যাডজাস্ট করা মানে একটা সংখ্যা সমাহার ভরে দেওয়া — ওই সংখ্যাগুলো আপাতত রয়ে যাচ্ছে ওই ভাঁড়ারে, যতক্ষণ-না নতুন করে অ্যাডজাস্ট করা হচ্ছে। একটা অংশ 'ইনপুট', পাঞ্চড কার্ড পড়ার, সেখান থেকে

আদেশ এবং সংখ্যা নেওয়ার। আর একটা অংশ 'আউটপুট' — সে প্রিন্টারেই হোক বা কার্ডে। গঠনটাকে এবার চেনা লাগতে শুরু করছে?

একটা এইটি কলাম বা আশি স্তম্ভের কার্ড — তিন ইঞ্চি বাই সাত ইঞ্চি , এতে আশি সারি তথ্য ঢোকানো যায়, ফুটো করে



করে। আমরা জানি কী করে ফুটো দিয়েই সংখ্যা, অক্ষর, চিহ্ন পাঠানো যায়। সেগুলো জুড়ে জুড়ে তথ্য — বিট থেকে বাইট থেকে নানা মাপের ওয়ার্ড। শুধু কম্পিউটারে কার্ডের ফুটো থেকে তথ্য পড়ার একটা কল থাকতে হবে।

অ্যানালিটিকাল ইঞ্জিনের এই পাঞ্চড কার্ড ব্যবহারের তরকিবটা এসেছিল জ্যাকার্ড লুম থেকে। জ্যাকার্ড লুম বা জ্যাকার্ডের তাঁত হল সেই প্রথম মেশিন যা পাঞ্চড কার্ড ব্যবহার করেছিল। ১৮০১ সালে ফরাসী উদ্ভাবক জোসেফ-মারি জ্যাকার্ডের তৈরি এই তাঁতে ২৪০০০ অব্দি শক্ত কার্ড লাগানো হত একটা ঘুরতে থাকা ড্রামে। যেখানেই কার্ডের গায়ে ফুটো আছে, সেখানেই একটা ছুঁচ ঢুকে গিয়ে পছন্দমত সুতো টেনে আনবে, আর ফুটো না-থাকলে ছুঁচটা নড়বে না। এভাবেই বোনা হবে কাপড়ের গায়ে নানা রঙের সুতোর আলপনা। সম্রাট নেপোলিয়ন এই মেশিনের জন্যে জ্যাকার্ডকে মেডেল দিয়েছিলেন। পরে এই পাঞ্চড কার্ডের আইডিয়া যায় ব্যাবেজের কাছে, এবং আরো পরে হেরমান হোলেরিথ-এর মেশিনে, যাতে স্ট্যাটিস্টিক্স-এর তথ্য নাড়াচাড়া করা হত। হেরমান হোলেরিথ-এর সেই

ব্যবসার কাজের মেশিন বানানোর ছোট্ট কোম্পানি, যার কাজ প্রথম জনমনযোগের পাদপ্রদীপে আসে মার্কিন সেন্সাসের প্রচুর তথ্য নাড়াচাড়ার কাজ দ্রুতবেগে করে দিয়ে, এই পাঞ্চড কার্ড ব্যবহার করে। ক্রমে বড় হতে থাকা এই কোম্পানির নাম ছিল ইন্টারন্যাশনাল বিজনেস মেশিনস। জানেন নাকি এই কোম্পানিটার কথা? শুধু নামের তিনটে শব্দের প্রথম বর্ণ দিয়ে পড়ে দেখুন তো একবার?

ব্যাবেজের অ্যানালিটিকাল ইঞ্জিন নিয়ে একটা লেখায় আডা লাভলেস লেখেন, জ্যাকার্ড লুম যেমন ফুটো করা কার্ড দিয়ে ফুল লতা পাতা ফোটায় কাপড়ের গায়ে, ঠিক তেমনি পাঞ্চড কার্ড দিয়ে অ্যালজেব্রার আল্পনা আঁকতে পারে ব্যাবেজের মেশিন। জ্যাকার্ড তাঁতের মত পাঞ্চড কার্ড আজো ব্যবহার হয় দেখবেন, সোয়েটার বোনার মেশিনে। এই পাঞ্চড কার্ডের কথা পরেও আসবে — এতে করেই কম্পিউটারে তথ্য ঢোকানো বার-করার কাজ চলেছে আধুনিক কম্পিউটারের গোড়ার দিকের প্রায় পুরো সময়টা জুড়েই। এতে বিটগুলো লেখা হয় সারিবদ্ধ ফুটো দিয়ে। এই ফুটো করার পদ্ধতির নাম হোলেরিথ কোডিং, হেরম্যান হোলেরিথ ১৮০০ সালে ফুটো কাজে লাগিয়ে এই তথ্য পাঠানোর পদ্ধতি তৈরি করেন। কার্ডটার মধ্যে দিয়ে বিদ্যুৎ যায়না, অপরিবাহী। যেখানে ফুটো সেখানে দুটো ধাতব অংশের মধ্যে সংযোগ ঘটছে, মানে বিদ্যুৎ যাচ্ছে, মানে তথ্য। হোলেরিথ এই ধরনের মেশিন বানানোর ব্যবসা খোলেন, যা, অনেক অনেক পরে, আজকের দুনিয়ার কম্পিউটার দৈত্য আইবিএম হয়ে দাঁড়িয়েছে। যাকগে, আমরা বেজায় রকমের কুপথে চলে যাচ্ছি, মানে ডাইগ্রেস করছি।

ব্যাবেজ এই জ্যাকার্ড লুমের ফুটো-করা কার্ডকে ব্যবহার করতে চাইলেন আরো অনেক সূক্ষ রকমে। তিনরকম ভাবে কার্ডের ব্যবহারের প্রস্তাব করলেন ব্যাবেজ।

'কার্ডস-অফ-দি-অপারেশনস', মানে কী করিতে হইবে তা যে কার্ডে লেখা থাকবে, যোগ না বিয়োগ না গুণ না ভাগ।

'কার্ডস-অফ-দি-ভ্যারিয়েবলস', মানে যাতে সেই স্তম্ভ বা কলামকে দেখানো হবে, যাদের শরীরের সংখ্যাগুলোকে করিতে হইবে, আর সেই চাকাগুলোকেও চিহ্নিত করা হবে, যেখানে গিয়ে এই করার ফলাফলটা পৌঁছবে।

'কার্ডস-অফ-দি-নাম্বারস', মানে, যাতে নির্দিষ্ট সব আঙ্কিক মান ভরা থাকবে, হিশেব করতে গিয়ে যাদের প্রয়োজন পড়ে। যেমন — পাই (π: ৩.১৪১৫৯২৬ . . .), ই (e: ২.৭১৮২৮১৮ . . .) ইত্যাদি।

আর, প্রত্যেকবারের প্রত্যেকটা কার্ডই কেবল তার নিজের জায়গায় ব্যবহার হবে তা নয়, কিছু নির্দিষ্ট কার্ডের এক একটা সমাহারকে বারবার করে ব্যবহার করা যেতে পারে, যেরকম আধুনিক প্রোগ্রামিং ভাষায় লুপ ব্যবহার হয়। এবং কার্ডের গায়ে খচিত সংখ্যার মাধ্যমে দেওয়া কোনো শর্তের সাপেক্ষে সম্পূর্ণ ভিন্ন কোনো কার্ডেও চলে যাওয়া যেতে পারে, মানে জাম্প। এই আলোচনায় বিশদ ভাবে আসছি আমরা, পরের সেকশনে, কম্পিউটার চিন্তনের ইতিহাসে।

ব্যাবেল বুঝেছিলেন যে শুধু হার্ডওয়ার নয়, তার ইঞ্জিন ঠিক ভাবে চালাতে গেলে একটা সফটওয়ারেরও প্রয়োজন পড়বে। এবং এই কাজে নিয়োগ করেছিলেন আডা লাভলেস নামে এক তরুণ কবিকে। আডা ছিলেন কবি বায়রনের একমাত্র বৈধ মেয়ে। পরে এই প্রথম সফটওয়ার অভিলাষিণীকে স্মরণ করে একটা কম্পিউটার ভাষার নাম রাখা হয়েছে আডা। আডা বলে এই কম্পিউটার ভাষাটা তৈরি করেছিল আমেরিকা সরকারের ডিপার্টমেন্ট অফ ডিফেন্স, সত্তরের দশকের শেষ দিকে, নতুনতর সফটওয়ার বানানোর উদ্দেশ্যে। এই জায়গাটায় ফের আমরা ফেরত আসব, আমাদের এই পাঠমালার শেষ মানে দশ নম্বর দিনে, ব্যাশ শেল প্রোগ্রামিং-এর প্রসঙ্গে কম্পিউটার ভাষার আলোচনায়।

১৮৩৩-এ ব্যাবেজ এই অ্যানালিটিকাল ইঞ্জিনের কাজ শুরু করার পর থেকে ১৮৪২ অব্দি ব্রিটিশ সরকার ব্যাবেজের জন্যে বরাদ্দ করেন তখনকার দিনের সতেরো হাজার পাউন্ড, আর ব্যাবেজ তার নিজের টাকা থেকে খরচ করেছিলেন বিশ হাজার পাউন্ড। তিতিবিরক্ত হয়ে তৎকালীন ব্রিটিশ প্রধানমন্ত্রী রবার্ট পিল জিগেশ করেছিলেন, মেশিনটা বানাতে আর কত সময় লাগবে হিশেব করার জন্যে ওই মেশিনটাকেই কাজে লাগালে হয়না?

আধুনিক কম্পিউটার প্রথম যারা বানান তাদের একজন হাওয়ার্ড আইকেন, চার নম্বর দিনে ভন নয়মান, আইকেন, জুসে, এবং এর পরবর্তী কম্পিউটার জেনারেশনের ইতিহাস — এইসব আসবে। এই আইকেনকে একবার প্রশ্ন করা হয়েছিল, কম্পিউটার উদ্ভাবনের তার এই কাজের প্রেরণা তিনি পেলেন কোথা থেকে? ব্যাবেজের বইটা দেখিয়ে আইকেন বলেন, 'এই আমার কম্পিউটার শিক্ষা, ঠিক এইটুকু — এটাই সব, আমি যা করেছি সব এই বই থেকে

নেওয়া।' চার্লস ব্যাবেজের ১৮৩২-এ লেখা বই, 'ইকনমি অফ মেশিন অ্যান্ড ম্যানুফ্যাকচার্স' থেকেই জন্ম নিয়েছিল আজকের দুনিয়ার খুব জরুরি একটা বিদ্যা — অপারেশনাল রিসার্চ।

চার্লস ব্যাবেজের সবচেয়ে গুরুত্বপূর্ণ উত্তরাধিকার হল সূক্ষ্ম যন্ত্রপাতি বা প্রেসিশন মেশিনিং এর এলাকাটায় — কী হবে এর বিভিন্ন যন্ত্রাংশের মান, গঠন, ইত্যাদি। গণিতের ইঞ্জিন বানাতে ব্যাবেজের প্রয়োজন ছিল অত্যন্ত উঁচু মানের সূক্ষ্ম যন্ত্রাংশ, সেই মুহূর্তের পৃথিবীর কোথাও যা সম্ভব ছিলনা। ব্যাবেজের অবদানটার গোটা নাটকীয়তাটা বোঝা যায়না ঠিক তার আগের শেষ অঙ্ক মেশিন কোলমারের অ্যারিথমোমিটারের সঙ্গে না মেলালে। ব্যাবেজ প্রথম চিন্তাপ্রক্রিয়াকে যন্ত্রে ধরার স্বপ্ন দেখালেন মানুষকে। ব্যাবেজের করা অ্যানালিটিকাল ইঞ্জিনের ডিজাইনগুলোর ভিত্তিতে ১৯৯১-এ, তার সময়ে যেসব জিনিষপত্তর পাওয়া যেত, তাই দিয়েই, নতুন করে বানানো হয়েছিল অ্যানালিটিকাল ইঞ্জিন, এবং তারা নিখুঁত ভাবে কাজ করেছিল — ঠিক ব্যাবেজ যেরকম বলেছিলেন। ২০০১-এ বানানো হয় এমনকি এর প্রিন্টিং ইউনিটটাও।

২।। কম্পিউটার চিন্তনের বদলের ইতিহাস

দ্বিতীয় বিশ্বযুদ্ধের আগে আগে আগে হাওয়ার্ড আইকেন, কনরাড জিউসে, ভন নয়ম্যান, উইলিয়ম মশলেদের ডিজিটাল কম্পিউটার থেকে পরবর্তী বদলগুলোয় আসব আমরা চার নম্বর দিনে। সেখানে মেশিনের ইতিহাস আর সেই মেশিনকে কাজ করানোর অপারেটিং সিস্টেমের ইতিহাস এক সঙ্গে জড়িয়ে, তাদের আলাদা করা যায় না। কিন্তু আজ যে হার্ডওয়ারগুলোকে চিনতে চিনতে আমরা এলাম সেগুলোয় কিন্তু সেই অর্থে কোনো সফটওয়ার নেই, মেশিনের সঙ্গে সফটওয়ারের জায়গাটা তৈরিই হয়নি তখনো। সেই তৈরি হওয়াটা একটা খুব বড় বদল, যুক্তির স্তরে, যুক্তিবোধের স্তরে, সেই যুক্তিবোধ দিয়ে বাস্তবতাকে চিন্তা করার স্তরে। তার পূর্বশর্তগুলো তৈরি হচ্ছিল ব্যাবেজের কাজ থেকে, বা আরো ভালো করে বললে, লাভলেস-ব্যাবেজের কাজ থেকে, বোধহয় সেখানে আগে লাভলেস, পরে ব্যাবেজ।

এবং, যা আমরা একাধিকবার আগেও বলেছি, একা কিছু হয়না, অনেক মানুষ নানা ভাবে চেষ্টা করতে থাকে, ইতিহাসের নানা পর্যায়ে, সেগুলোর মধ্যে আপাতগোপন কিছু সময়ের ছক রয়ে যায়, হঠাৎ একজনের চোখে পড়ে সেটা, সে তখন সেই লাইনে এগোয়, তার এগোনোটা তার সময়ের মধ্যে সীমাবদ্ধ। তার কাজের মধ্যে রয়ে যায় আবার নতুনতর কিছু ছকে মিলে যাওয়ার সম্ভাবনা। অনেক মানুষ এবং অনেক সময় মিলে কাজগুলো করে। এই অনেক মানুষের মধ্যে মধ্যে থেকে আরো দুজনকে আমরা বেছে নেব, জর্জ বুলি আর অ্যালান টুরিং, আমাদের এই কম্পিউটার চিন্তনের ইতিহাসের আলোচনায়। বাদ পড়বেন আরো বহু মানুষ, বাদ পড়বে আরো বহু সময়, যার পুরোটার এমনকি ইতিহাসও নেই। বাদ পড়বেন ভন নয়ম্যান, নরবার্ট ওয়েইনার যারা প্রথম যুগের কম্পিউটার চিন্তাকে গজিয়ে তুলেছিলেন, বাদ পড়বেন কম্পিউটার জগত থেকে স্বেচ্ছানির্বাসিত বার্নারস লি, আজকের আমাদের কম্পিউটারময় জীবনের পিছনে রয়েছে যাদের শ্রম ও স্বপ্ন, যে স্বপ্নের অনেকটাই ধর্ষিত হয়েছে মাইক্রোসফট ও অন্যান্য আন্তর্জাতিক পুঁজির হাতে। যাই হোক, আসা যাক লাভলেস ও ব্যাবেজের কথায়।

২.১।। প্রথম প্রোগ্রামার আডা লাভলেস এবং রগচটা প্রতিভা চার্লস ব্যাবেজ

চার্লস ব্যাবেজকে এই রগচটা প্রতিভার অভিধা, 'ইর্‌্যাসকিবল জিনিয়াস', দিয়েছিলেন ব্যাবেজের এক জীবনীকার। ব্যাবেজের রগচটা ক্রোধ ছিল প্রবাদপ্রতিম। ব্যাবেজ ছিলেন অভিজাত উচ্চবিত্ত ঘরের থরোব্রেড ইংরেজ। তার এই বিত্ত গোটাটাই গেছিল ওই অসমাপ্ত উদ্ভাবনে। ব্যাবেজের উদ্ভাবন কিন্তু ওটাই প্রথম নয়। এর আগে কাউক্যাচার বানিয়েছিলেন ব্যাবেজ। স্টিম-ইঞ্জিনের ট্রেনের সামনে যা লাগানো হত, পথভোলা দিওয়ানা গরুরা যাতে কোনো দুর্ঘটনায় না-পড়ে, তখনকার ইংলন্ডে ওদের তো কোনো জীবনবীমা থাকত না।

গোটা একটা শিল্প বা ইন্ডাস্ট্রিকে, তার উৎপাদন প্রক্রিয়াকে বিশ্লেষণ করার উপায় উদ্ভাবন করেন ব্যাবেজ, আগেই বলেছি, অপারেশনাল রিসার্চ। তার এই বিশ্লেষণ প্রক্রিয়া তিনি কাজে লাগান মুদ্রণ-শিল্পের উপরেই, তার ফলাফল এই হয় যে, ব্যাবেজের নিজেরই পাবলিশার চটে গিয়ে ব্যাবেজের বই ছাপা বন্ধ করে দেয়। ব্যাবেজই দেখান, পোস্টে কোনো চিঠি পাঠানোর সময়, কতটা দূরত্ব চিঠিটা যাবে সেই অনুযায়ী দাম ঠিক করার জন্যে যা খরচ হয় তা পাঠানোর

খরচের চেয়ে বেশি। ব্রিটিশ পোস্টব্যবস্থা সঙ্গে সঙ্গে নিয়মটা বদলে সব দূরত্বের জন্যে একই দাম করে দেয়, গোটা পৃথিবী জুড়েই এখনো যা চালু।

রেলপথের প্রথম স্পিডোমিটার বানান ব্যাবেজ। ইনশিওরেন্স বা বীমার রিস্ক আর প্রিমিয়াম, ঝুঁকি আর দেয়, নিয়ে বই লেখেন, যা থেকে বীমা কোম্পানিগুলো প্রচুর লাভ করে। সাইফারিং ডিসাইফারিং করতে, সঙ্কেত বানাতে এবং পাঠোদ্ধার করতে জানতেন ব্যাবেজ। এবং পরবর্তীকালের কম্পিউটার এনক্রিপশন বা সঙ্কেতীকরণের ভাবনার কিছু জায়গা এসেছে তার কাজ থেকেই। গাছের গুঁড়ির মধ্যে পরপর রিং গুলোয় যে আবহাওয়া বদলের ইতিহাস লিপিবদ্ধ থাকে এটাও তিনিই প্রথম বলেন। এগুলো গেল কেজো আইডিয়া, ব্যাবেজের প্রচুর আইডিয়া ছিল গুংগা মেশিনের মতই বিদঘুটে। তাদের বিদঘুটেপনা ইতিহাসের হাতে প্রমাণিততর হয়েছে। ব্যক্তিগত জীবন এবং সামাজিক সম্পর্কে ব্যাবেজ ছিলেন ওইসব আইডিয়ার মতই বিদঘুটে এবং উৎকেন্দ্রিক। লন্ডনের রাস্তার ভ্রাম্যমান অর্গান বাজিয়েদের বিরুদ্ধে পত্রিকায় চিঠি লিখতেন ব্যাবেজ, বিপরীতে তারাও এসে বাজনা ঘাড়ে সদলে জড় হত ব্যাবেজের জানলার নিচে।

১৮২২-এ ব্যাবেজ রয়াল অ্যাস্ট্রোনমিকাল সোসাইটির সভায় একটা ছোট মডেলের সক্রিয় ডিফারেন্স ইঞ্জিন দেখান ব্যাবেজ, সোসাইটি তাকে মেডেল দেয়। পরের বছর থেকে তার গ্রান্ট আসা শুরু হয়। ব্রিটিশ সরকার যে ডিফারেন্স ইঞ্জিনের অর্ডার দেয় সেটার আকার ছিল বড্ড বেশি বড়। মূলত টেকনোলজির প্রতিবন্ধকতায় ব্যাবেজের কাজ শেষ হয়না। নানা লোকে ব্যঙ্গবিদ্রূপ করা শুরু করে দেয়, পাগলা বুড়ো চার্লি ব্যাবেজ, ইত্যাদি। এইসবের বিরুদ্ধে, সরকারের গ্রান্ট বন্ধ করে দেওয়ার বিরুদ্ধে লড়াইটা চলছে, এই অবস্থায় এই সময়ে তার মাথায় আসে পরবর্তী আইডিয়া — অ্যানালিটিকাল ইঞ্জিন, তার কাছে মহত্তর, অন্যের কাছে উন্মাদতর। ব্যাবেজের যুক্তিটা ছিল এই যে, এক ধরনের একটা হিশেব যদি মেশিনে করা যায় তাহলে যে কোনো জটিলতার যে কোনো হিশেব কেন মেশিন দিয়ে করা যাবেনা? বিরাট বিরাট জাম্বো হিশেব? অনেক রকমের অনেক হিশেব করার জন্যে আলাদা আলাদা মেশিন বানাব কেন? একটা বড় মেশিন বানাব, তার নানা অংশগুলো নানা সমাহারে এসে নানা ধরনের হিশেব করবে। ব্যাবেজের এই সর্বব্যাপী হিশেবকল — ইউনিভার্সাল ক্যালকুলেটিং মেশিনের ধারণাটা এসে বিস্ফোরিত হয়েছিল অনেক পরে অ্যালান টুরিং-এর চিন্তাকাঠামোয়।

‘মিল’, ‘স্টোর’ এবং ‘কার্ড’ দিয়ে তৈরি ব্যাবেজের এই অ্যানালিটিকাল ইঞ্জিনের ছকটা তো আমরা আগেই বলেছি। মিল বা হিশেবক দশমিকের পর মাত্র পঞ্চাশ ঘর পর্যন্ত হিশেব করবে, দুর্ধর্ষ গতিতে এবং একটাও ভুল ছাড়া — গ্যারান্টি। আর স্টোর বা ভাঁড়ার এক হাজার খানা পঞ্চাশ অঙ্কের সংখ্যা ধরে রাখতে পারবে তার ডান্ডা এবং গিয়ার লাগানো উদরে। ফুটোসজ্জিত কার্ডমালা সংখ্যা জুগিয়ে যাবে ক্ষুধার্ত ভাঁড়ারকে। ফলাফল বেরিয়ে আসবে স্বয়ংক্রিয় ছাপার যন্ত্রে।

শুধু তাই নয়, দরকার পড়লে কার্ড দিয়ে এই আদেশও দেওয়া যাবে মিলকে যে তুমি আপাতত কিছু সংখ্যা তুলে রাখো ভাঁড়ারের তাকে, পরে হিশেবের কাজে প্রয়োজনমত ফেরত এনো। অর্থাৎ কার্ড পড়ার যন্ত্র বা কার্ড-রিডিং-ডিভাইসটাই এখানে নিয়ন্ত্রক এবং সিদ্ধান্তকারী সংস্থা, কনট্রোল এবং ডিসিশন-মেকিং ইউনিট হয়ে কাজ করবে। আমাদের শূন্য নম্বর দিনের কাজের ভিত্তিতে একটা কম্পিউটারের ছকের সঙ্গে তুলনা করুন — আমরা একে খাপে খাপে মিলিয়ে ফেলতে পারছি আজকের আধুনিক একটা পিসির কাজের ছকের সঙ্গে। মিল এখানে সিপিইউ। স্টোর হল বিভিন্ন রকমের মেমরি উপাদান। কার্ড হল ইনপুট ডিভাইস আর কন্ট্রোল ইউনিট। আর প্রিন্টার হল আউটপুট ডিভাইস। এই কার্ড নামক ইনপুট এবং কন্ট্রোল ডিভাইসের কারণেই এই অ্যানালিটিকাল ইঞ্জিন কম্পিউটারের প্রাগইতিহাস থেকে বেরিয়ে এল আধুনিক কম্পিউটারের সঙ্গে আত্মীয়তার গৌরবে।

ব্যাবেজের মেশিনের এই ইনপুট সংস্থা কম্পিউটার তথা প্রোগ্রামিং-এর ইতিহাসে একটা উজ্জ্বল মাইলস্টোন। পাঞ্চড কার্ডের আইডিয়া এসেছিল জ্যাকার্ড লুম থেকে, আমরা আগেই বলেছি। জ্যাকার্ড তাঁতে সুতো টানার জন্যে ব্যবহার হত ধাতুর তৈরি শক্ত ছুঁচ, আর অলঙ্করণ বানানোর ছকটা দেওয়া হত শক্ত কার্ড দিয়ে, যে কার্ডে এমনিতে ছুঁচগুলো আটকে যাবে, ঢুকতে পারবে একমাত্র ফুটো থাকলেই। তাঁতের প্রত্যেকবার ঘোরায় ছুঁচাবলীর সামনে আসবে একটা নতুনতর কার্ড। স্বয়ংক্রিয় হিশেবের পরিকল্পনা ব্যাবেজের মাথায় এল এই ফুটো করা কার্ড থেকেই। গাণিতিক চিন্তন

প্রক্রিয়ার বিমূর্ত অবয়বকে মূর্ত বস্তুতে হাজির করার এই প্রথম কোনো উপায় পাওয়া গেল। একটা জটিল হিশেবের সম্পূর্ণ সম্ভাব্য আকারটাকে ভেঙে ফেলা হল ছোট ছোট পদক্ষেপে — একটা ফ্লো-চার্ট বা প্রবাহ তালিকায়।

ফ্লো-চার্ট হল সময়ানুক্রমিক ভাবে পরপর আসার ছোট ছোট টুকরোয় ভেঙে ফেলা একটা ক্রিয়াপদ্ধতি। প্রোগ্রাম করার আগে কী করতে চাইছি — গোটা কাজের প্রোজেক্টটাকে এরকম ফ্লো-চার্টে ভেঙে ফেলা হয়। কিন্তু ফ্লো-চার্ট করা যেতে পারে যে কোনো কাজেরই। একটা বড় কোনো কাজকে ছোট ছোট সহজবোধ্য সরলবোধ্য ছোট ছোট কাজলেট বা কুচো কাজে ভেঙে ফেলা, যা এমনকি বুদ্ধিকুলতিলক একটা রোবট বা কম্পিউটারও পরপর করে যেতে পারবে। প্রোগ্রামিং শেখার শুরুতে ছাত্রদের ফ্লো-চার্ট বানাতে হয় — ফ্লো-চার্ট ব্যাপারটা এই কাজের তালিকাটাকে একটা ছবি দিয়ে বোঝায়।

কার্ডের পর কার্ডে এবার দেগে দেওয়া হল ফ্লো-চার্টটাকে। এই কার্ডগুলো এবার পরপর অ্যানালিটিকাল ইঞ্জিনের মিল মানে সিপিইউ-টাকে নিয়ন্ত্রণ করবে, একদল বাধ্য সৈনিকের মত। কাজটা বদলানো দরকার, ঠিক আছে, আর এক সেট কার্ড আনো, বদলে ফেলো সৈনিকদের, যারা পাহারা দিয়ে চলেছে মিলকে, ঠিক ঠিক জায়গায় ঠিক ঠিক কাজ করাচ্ছে। পুরো প্রক্রিয়াটাকে নিজের মনের আয়নায় খুব ভালো ভাবেই দেখতে পেয়েছিলেন ব্যাবেজ, কিন্তু সেটা কাঠ আর লোহা আর পিতলের কলকব্জায় সঠিক ভাবে অনুবাদ করতে পারেননি শত চেষ্টাতেও, আগেই তো বললাম, শতাব্দীটা ভুল ছিল তার জন্মানোর এবং কাজ করার পক্ষে।

আডা লাভলেস

কিন্তু অন্তত একজনকে পেয়েছিলেন ব্যাবেজ যে এই পুরো ব্যাপারটা তারই মত করে, বা হয়তো তার চেয়েও বেশি দার্শনিক গভীরতায় দেখতে পেয়েছিল। আডার সঙ্গে ব্যাবেজের এবং তার ইঞ্জিনের যখন প্রথম মোলাকাত হয় আডা লাভলেস তখনো কুমারী আডা বায়রন। আডার বাবা কবি বায়রন ছিলেন তার সময়ের ইংলন্ডের সবচেয়ে কেচ্ছাতাড়িত মানুষদের একজন। লেডি বায়রন অভিযোগ করেন বায়রনের নিজের এক বোনের সঙ্গে বায়রনের যৌনতার, পরবর্তী ঘটনাধারা যার সম্ভাব্য সত্যতার দিকেই ইঙ্গিত করে, সেই বোনের যে সন্তান হয় তার সঙ্গে আডার মিলের কথা উল্লেখ করে এই মন্তব্য করেছেন আডার এক জীবনীকার। আডার জন্মের একমাস পর থেকে বায়রন কখনো আর আডার মার কাছে আসেননি। কিন্তু তার একাধিক কবিতায় আডার উল্লেখ আছে। আডার মা, তার সময়কার অন্যান্য স্বাধিকারপ্রমত্তা ভিক্টোরিয়ানীদের মতই আর একজন। নিজের সুন্দরী কিন্তু ঠোঁটকাটা, প্রতিভাবান কিন্তু দলছুট কন্যা আডাকে সামলানোর মত ব্যক্তিত্বের অভাবেই হয়ত, লেডি বায়রন ছোটবেলা থেকেই আডাকে অভ্যস্ত করেন লডেনাম বা আফিমজাত ড্রাগ মেশানো টনিকে, যে নির্ভরতার থেকে আডা তার জীবনের ছত্রিশ বছরে বেরিয়ে আসতে পারেনি। আডার মা বেঁচে ছিলেন মেয়ের মৃত্যুর পরেও।

খুব ছোটবেলা থেকেই আডার গাণিতিক প্রতিভার কমতি ছিলনা। ঘনিষ্ঠ পারিবারিক বন্ধুদের একজন ছিলেন অগাস্টাস ডি মরগান, জর্জ বুলির পাশাপাশি আধুনিক সিম্বলিক লজিকের বা চিহ্ননির্ভর যুক্তিবিজ্ঞানের আর একজন

সহ-জন্মদাতা। আডার গণিতশিক্ষার কিছুটা এই ডি-মরগানের কাছে। তারপর যা স্ফূরণ পেতে শুরু করে ব্যাবেজের সান্নিধ্যে। তখনকার ইংলন্ডে চল ছিল নানা নতুন যন্ত্র এনে অভিজাতদের পার্টিতে একটা প্রদর্শনী দেওয়া। শিল্পবিপ্লবের ব্রিটেনে সবচেয়ে বড় প্রতিমা বা ফেটিশ তখন যন্ত্র। এই রকমই একটা পার্টিতে হাজির ছিলেন অগাস্টাস ডি মরগানের স্ত্রী শ্রীমতী ডি মরগান এবং আডা, যেখানে ব্যাবেজ তার ডিফারেন্স ইঞ্জিন নিয়ে আসেন। পরে তার স্মৃতিচারণে শ্রীমতী ডি মরগান লিখেছেন —

> একটা আয়না দেখে বা বন্দুকের আওয়াজ শুনে জংলিরা যেমন করে তাকায় বলে শুনেছি, অন্য সবাই যখন সেইরকম করে ওই চমৎকার উদ্ভাবনটা দেখছিল, আমাদের কুমারী বায়রন তখন, আরো ছোট তো, লেগে গেল বুঝতে — কেমন করে কাজ করছে, আর আবিষ্কারটার সত্যিকারের সৌন্দর্যটা দেখল (While the rest of the party gazed at this beautiful invention with the same sort of expression and feeling that some savages are said to have shown on first seeing a looking glass or hearing a gun, Miss Byron, young as she was, understood its working and saw the great beauty of this invention.)।

এই উদ্ধৃতিটা অবশ্যই প্রতিভাবান কিশোরী আডার ব্যক্তিস্বাতন্ত্র্যকে চিহ্নিত করে। কিন্তু আমি অনুবাদ করলাম আরো একটা মজায়। আজ থেকে মোটামুটি ১৭৫ বছর আগে, তার মানে প্রচলিত পঁচিশ বছর হারে সাতটা জেনারেশন, অর্থাৎ, নিজের চারপাশে বলয়ে বলয়ে পরিবর্দ্ধমান ঔপনিবেশিক ইংলন্ডের ক্রমগজায়মান কলোনি নেটিভ ভারতে, ওই 'জংলি' বা 'স্যাভেজ' অভিধায় ধরা হচ্ছে আমার ঠাকুর্দার ঠাকুর্দার ঠাকুর্দার পনেরো বয়স্ক ঠাকুর্দাকে বা তার চল্লিশ বয়স্ক বাবাকেও। মালেরা কি হেভি চমকাত বন্দুকের শব্দে? আমি তো যদ্দূর জানি, আমার একটা সুস্পষ্ট লেঠেল পাইক খুনোখুনির জিনিলজি আছে, একটা বল্লমও দেখেছি আগে, ছোটবেলায়, আমার সাংস্কৃতিক উত্তরাধিকার, পুরো সালা-লুট-লিয়া বল্লম। সেই বল্লমের থেকে বন্দুকের জোরের এফিসিয়েন্সিতেই ইংরেজরা বোধহয় লেঠেলতর করে নিজেদের প্রতিষ্ঠিত করে গেল। অথচ দেখুন, অতীশ দীপঙ্কর চীন থেকে শুধু বৌদ্ধশাস্ত্ররাশিই আনেননি, এনেছিলেন বারুদ বানানোর টেকনোলজিও। যাকগে।

অন্যদের কাছে এই ক্যালকুলেটর আর একটা প্রদর্শনীর জলের পাম্পের থেকে আলাদা কিছু ছিলনা, কিন্তু আডাকে এটা চূড়ান্তভাবে উত্তেজিত করল। আডা বায়রন — পৃথিবীর প্রথম কম্পিউটার হুইজ কিড। আডা সেই হাতে গোনা কয়েকজনের একজন যারা ব্যাবেজের এই ইঞ্জিনকে এর আগের প্রজন্মের যান্ত্রিক ক্যালকুলেটরগুলোর থেকে সম্পূর্ণ ভিন্ন কিছু বলে চিনতে পেরেছিল। আগের প্রত্যেকটা যান্ত্রিক ক্যালকুলেটরই ছিল অ্যানালগ, যারা পরিমাপ দিয়ে কাজ করে, কিন্তু ব্যাবেজের এই মেশিন হল ডিজিটাল, হিশেব করছে গুনে গুনে, এক দুই তিন করে। অন্য অ্যানালগদের কাছে যেখানে হয়ত ৩৬ ডিগ্রি বা এক ইঞ্চি মানে এক, ৭২ ডিগ্রি বা দুই ইঞ্চি মানে দুই, ইত্যাদি। সবসময়ই একটা ভৌত-মান আর তার সংখ্যা-মানের ভিতর অনুবাদ-প্রতিবাদ করে চলতে হয় অ্যানালগদের। তার চেয়েও বড় কথা ব্যাবেজের এই মেশিনই প্রথম যেখানে গণিতের আর যুক্তিবিজ্ঞানের, অ্যারিথমেটিক আর লজিক, দুই ধরনের কাজই এক সঙ্গে আনা হল। মিল করছে পাটীগণিত আর কার্ড করছে যুক্তিবিজ্ঞান।

ব্যাবেজ তার জীবনের শেষ দিকে সদ্য জন্মানো বুলি আর ডি মরগানের লজিক হাতে পেয়েছিলেন, কিন্তু আডা তখন আর বেঁচে নেই। লেডি অগাস্টা আডা বায়রন কাউন্টেস অভ লাভলেসের জীবনটা ছিল সংক্ষিপ্ত, ছত্রিশ বছরের, ১৮১৫-র দশই ডিসেম্বর থেকে ১৮৫২-র সাতাশে নভেম্বর। ঠিক যে বয়সে তার বাবা বায়রনও মারা গেছিলেন, দূরে পরবাসে, গ্রীসে। প্রফেসর মরিয়ার্টির কথা বলতে গিয়ে শার্লক বলেছিল, রক্তে রয়ে যাওয়া শিল্প কখন কী আকার নেবে বলা যায়না, আর্ট ইন ব্লাড টেকস কিউরিয়াস টার্নস, বাবা বায়রনের উত্তরাধিকার তেমনি আডাকে কখনো শান্ত থাকতে দেয়নি। বাজনা বা ক্যালকুলাস, কখনো তীব্র সক্রিয়তা, কখনো চূড়ান্ত ডিপ্রেশন। এই নিরবচ্ছিন্নভাবে অসুস্থতাক্রান্ত সংক্ষিপ্ত অথচ ঘটনাবহুল জীবনে আডাকে কিছুটা সময় শিলন-এর সুইস জেলেও কাটাতে হয়েছে।

অ্যানালিটিকাল ইঞ্জিন বা এই ধরনের লজিক ডিভাইস নিয়ে ঠিক কী কী করা যেতে পারে তাই নিয়ে আডার চিন্তা ছিল ব্যাবেজের থেকেও স্বতন্ত্র। আডার লজিক শিক্ষার হাতেখড়ি ডি মরগানের কাছে। এবং এই ধরণের আধা গাণিতিক আধা যুক্তিবৈজ্ঞানিক ক্রিয়াকলাপে আডার ওস্তাদি এতটাই ছিল যে ব্যাবেজ লিখেছিলেন, "এটা ও আমার চেয়েও ভালো বোঝে, এবং আমার চেয়ে বহু বহু গুণ ভালো বোঝাতে পারে"।

উনিশ বছর বয়সে আডা বিয়ে করেন ব্যারন অফ লাভলেসকে, তিনিও গণিতজ্ঞ, তবে আডার দরের নন। গাণিতিক ইঞ্জিন নিয়ে ব্যাবেজ আর আডার যৌথ প্রকল্প চলতেই থাকে। ব্রিটিশ ক্ষমতার অপছন্দের সামনে দাঁড়িয়েও ব্যাবেজের কাজে আডা তার শর্তহীন সমর্থন জানাতে কখনো কসুর করেননি। মূলত ব্যাবেজের উৎসাহেই অ্যানালিটিকাল ইঞ্জিন সংক্রান্ত নিজের গোটা নোটসটা আডা প্রকাশ করেন। এবং সেই নোটস পুরোটাই, এতদিন পরেও, যৌক্তিক ভাবে সম্পূর্ণ বোধ্য। এবং প্রোগ্রামারদের কাছে অবশ্যই মূল্যবান।

পরিষ্কার বোঝা যায়, অ্যানালিটিকাল ইঞ্জিনের এই তাত্ত্বিক তার নিজের সময়ের থেকে কতটা এগিয়ে ছিলেন। ম্যাথামেটিকাল গেজেটে আডার নোটস নিয়ে নিউম্যান মন্তব্য করেন যে এই নোটসগুলোই প্রমাণ, আডা লাভলেস একটা প্রোগ্রামড কম্পিউটারের প্রতিটি নীতিই বুঝতে পেরেছিলেন, কম্পিউটারটা ঘটে ওঠার প্রায় একশো বছর আগে। ব্যাবেজের মেশিনে তথ্য আর সমীকরণ ঢোকানোর ওই পাঞ্চড কার্ডের কাজ নিয়ে বিশেষ ভাবে মনোযোগী ছিলেন আডা। তার 'অবজার্ভেশন অন মিস্টার ব্যাবেজস অ্যানালিটিকাল ইঞ্জিন' নামে একটা লেখা থেকে একটা প্যারা অনুবাদ করা যাক।

> কার্ড লাগানোর কথা মাথায় আসা মাত্র, পাটীগণিতের সীমানাগুলো ভেঙে বেরিয়ে গেল, অ্যানালিটিকাল ইঞ্জিন আর নিছক 'ক্যালকুলেটিং মেশিন'-গুলোর সঙ্গে একই জমিতে দাঁড়িয়ে রইল না। এর নিজেরই একটা অবস্থান হল, এবং এই অবস্থানে দাঁড়িয়ে যে যে ধারণাকে জাগিয়ে তুলল সেগুলো চূড়ান্তভাবে চিত্তাকর্ষক। *সাধারণ* চিহ্নগুলোকে (*জেনেরাল* সিম্বল) একত্রে নিয়ে আসার কৌশলটা, তার অন্তহীন বৈচিত্র এবং প্রসার নিয়ে, একটা ঐক্যের বন্ধন তৈরি করল পদার্থের ক্রিয়া আর বিমূর্ততম গাণিতিক বিজ্ঞানের বিমূর্ত মানসিক প্রক্রিয়ার মধ্যে। গাণিতিক বিশ্লেষণের (*অ্যানালিসিস*) ভবিষ্যত ব্যবহারের একটা নতুন, প্রসারিত এবং শক্তিশালী ভাষা জন্ম নিল, যা দিয়ে সত্যের চর্চা করা যাবে, যাতে সেটা এতটাই দ্রুত এবং ত্রুটিহীন রকমে মানবজাতির বাস্তব প্রয়োগে আসতে পারবে যা এর আগে কখনো ভাবা যায়নি। তাই শুধু মানসিক এবং প্রাকৃতিক নয়, গণিত জগতে যা তাত্ত্বিক এবং প্রায়োগিক তারা এবার এ অন্যের ঘনিষ্ঠ সান্নিধ্যে এল। অ্যানালিটিকাল ইঞ্জিনকে যা স্পষ্ট ভাবে চিহ্নিত করছে তার তুল্যমূল্য কোনো ভূমিকা অন্য কোনো কিছুর বেলায় প্রস্তাবিত হয়েছে, বা এমনকি কোনো বাস্তবিক সম্ভাবনা হিশেবে একটা চিন্তা-করা যুক্তি-বদ্ধ কোনো কিছুকে ভাবাও হয়েছে — এরকম কোনো তথ্য আমাদের গোচরে নেই।

> The bounds of *arithmetic* were, however, outstepped the moment the idea of applying cards had occurred; and the Analytical Engine does not occupy common ground with mere "calculating machines." It holds a position wholly its own; and the considerations it suggests are most interesting in their nature. In enabling mechanism to combine together *general* symbols, in successions of unlimited variety and extent, a uniting link is established between the operations of matter and the abstract mental processes of the *most abstract* branch of mathematical science. A new, a vast and a powerful language is developed for the future use of analysis, in which to wield its truths so that these may become of more speedy and accurate practical application for the purposes of mankind than the means hitherto in our possession have rendered possible. Thus not only the mental and the material, but the theoretical and the practical in the mathematical world, are brought into intimate connexion with each other. We are not aware of its being on record that anything partaking of the nature of what is so well designated the *Analytical* Engine has been hitherto proposed, or even thought of, as a practical possibility, any more than the idea of a thinking or a reasoning machine.

আমি জানিনা, আপনারা উত্তেজিত হচ্ছেন কিনা, আমি হয়েছিলাম, এই প্যারাটা পড়ে। প্যারাটা দেখুন, শুধু নামটা বদলে দিয়ে লাগিয়ে দেওয়া যেত ডিজিটাল কম্পিউটারের কোনো প্রারম্ভিক বর্ণনা বলে। এবং, পড়ামাত্রই আমার মাথায় এলো, একশো বছরেরও বেশি আগে লেখা এই প্যারাটার চেয়ে বড় সায়েন্স ফিকশন আর কী হতে পারে। ভাবুন, কম্পিউটারের গোটা ধারণাটা, মানবজীবনের উপরে তার প্রভাবিত বদলের নাটকীয়তাটা এতটাই বিচিত্র এবং আগের অভিজ্ঞতার থেকে ভিন্ন যে কোনো একটা সায়েন্স ফিকশনেও তো আমি কখনো কোনো ভবিষ্যদ্বানী পাইনি কম্পিউটারের। থমাস মোরের ইউটোপিয়া বাদ দিন, ওটা সাহিত্যের চেয়ে দর্শন বলা ভালো। গোড়ার দিকের এডগার অ্যালান পো বা মার্ক টোয়েন বা রুডিয়ার্ড কিপলিং থেকে শুরু করে জুল ভের্ন, এইচ জি ওয়েলস, এডগার রাইস বারোজ, আর্থার কোনান ডয়েল, আলডুস হাক্সলে অব্দি আমি একটা লেখাও মাথায় আনতে পারছিনা, বা

আরো পরের ব্রিলিয়ান্ট রুশ লেখক ভ্যালেন্টিলা ঝুরাভলিওভা অব্দিও, কেউ কম্পিউটার আসার আগে কম্পিউটারকে লিখেছেন। রে ব্রাডবেরির পুরোনোতম লেখা আমি যা পেয়েছি, ফারেনহেইট, ১৯৫৩-র লেখা, তখন কম্পিউটার এসে গেছে, গৌরবের দিকে এগোচ্ছে। রে ব্রাডবেরির মত যাদুকরের আস্তিনে আরো কিছু থাকতে পারে — আমি জানিনা। আপনাদের যদি এইরকম কোনো সায়েন্স ফিকশন মাথায় আসে একটু জানাবেন।

গণিতজ্ঞ আডার এতে উৎফুল্ল হওয়ারই কথা যে বড় বড় পরিশ্রমসাধ্য কষে যাওয়ার কাজগুলো এবার মেশিনই করতে পারবে। কিন্তু, আডার বেলায়, তার চেয়েও বড় গতিটা তৈরি হল এই ইঞ্জিনের প্রোগ্রামিং সংক্রান্ত নীতিগুলোকে তাদের চূড়ান্ত জায়গা অব্দি বোঝার চেষ্টা করা। যদি ওই বয়সেই মারা না যেতেন, আডা বোধহয়, সেই উনবিংশ শতাব্দীর প্রথমার্ধে দাঁড়িয়েই সত্যিকারের প্রোগ্রামিং ল্যাংগুয়েজ-এ পৌঁছে যেতে পারতেন। ইঞ্জিনটা তখনো তৈরিই হয়নি, আডা ওই কার্ডে প্রদেয় আদেশ-তালিকা বা ইন্সট্রাকশন সিকোয়েন্সগুলো ঠিক কী রকম হতে পারে তাই নিয়ে নানা পরীক্ষানিরীক্ষা শুরু করে দিয়েছিলেন। আবিষ্কার করেছিলেন এমন কিছু কায়দা যা আজকের কম্পিউটার ভাষাগুলোতেও অত্যন্ত প্রাথমিক এবং জরুরি জায়গা। সাবরুটিন, লুপ, জাম্প। এই প্রসঙ্গটা একটু মাথায় রাখবেন, এখানটায় বেশ বড় করে আমরা ফেরত আসব আমাদের পাঠমালার শেষ মানে দশ নম্বর দিনে।

ধরুন অনেক ছোট ছোট সরলতর হিশেবকে একত্রে সমাহৃত করে, বুনে বুনে, গড়ে তোলা হচ্ছে একটা জটিল এবং বড় হিশেব। এবার এই ছোট অংশগুলোর অনেকগুলোই বারবার আসবে, একই ধরনের হিশেব, একই রকম কিছু পদক্ষেপ, এক একটা গুচ্ছে, বারবার আসতে থাকবে। বারবার প্রত্যেকবার আপনাকে মোট বড় হিশেবটার মধ্যে লিখে যেতে হবে সেগুলো, ছোট ছোট স্টেপগুলোর সেই সমাহারটা, স্টেপগুলোর সংখ্যা বারোটাও হতে পারে আবার একশো চুয়াল্লিশটাও হতে পারে। তার চেয়ে এই বারবার একই আকারে ব্যবহৃত স্টেপগুচ্ছকে কেন বাপু একবার একটা জায়গায় লিখে ফেলো না? — একটা সাবরুটিনে? এইরকম সব সাবরুটিনরা সব এক সঙ্গে থাকবে একটা লাইব্রেরিতে, প্রয়োজন পড়লেই আমাদের কম্পিউটার প্রোগ্রাম যে লাইব্রেরি থেকে যে কোনো সাবরুটিনকে ডেকে নিতে পারবে, যতবার খুশি, যেখানে খুশি। এক নম্বর দিনে আমরা যে সি প্রোগ্রামটা লিখেছিলাম, তার স্ট্যান্ডার্ড ইনপুট আউটপুট হেডার 'stdio.h' ফাইলটা এইরকম একটা লাইব্রেরি ফাংশন, তবে নিছক সাবরুটিনের অর্থে নয়, ওখানে হার্ডওয়ার কী ভাবে ব্যবহার করবে সেই সিস্টেম-কলের ব্যাপারটাও ছিল। আজকের আমাদের ব্যবহারের প্রতিটি কম্পিউটার ল্যাংগুয়েজেই এই রকম এক একটা লাইব্রেরি থাকে।

অ্যানালিটিকাল ইঞ্জিন খুব দ্রুত প্রচুর প্রচুর হিশেব বারবার করেই যেতে, করেই যেতে, পারত। সেই করেই চলার আদেশগুলো অ্যানালিটিকাল ইঞ্জিনে দেওয়া শুরু হল কার্ড দিয়ে। যেই কার্ড দিয়ে আদেশ দেওয়া শুরু হল, তখনই, খেয়াল করুন, একটা সম্ভাবনা তৈরি হয়ে যাচ্ছে কোনো একটা বিশেষ আদেশের কার্ডে ফেরত যাওয়ার। কারণ সেই আদেশের কার্ডটা তো মেশিনে ভরাই আছে। এবার, এটা চিন্তার মাত্র আর একটা মোচড় যে, আর একটা কার্ড দিয়ে আদেশ দেওয়া যায় যে তুমি ওই বিশেষ কার্ডটায় ফেরত যাও। যেই এই ব্যবস্থাটা উদ্ভাবিত হল যাতে একটা কার্ড মেশিনকে আদেশ দিতে পারবে আগের কোনো একটা কার্ডে ফেরত যেতে, তারপর ফের পরপর কার্ড অনুযায়ী কাজ করে যাবে মেশিন — তৈরি হল লুপ। আবার চলতে চলতে ফের আসবে সেই কার্ডে যা তাকে ফেরত পাঠিয়েছিল, আবার ফেরত যাবে পুরোনো কার্ডে। এই ভাবে চলতেই থাকবে যতক্ষণ অব্দি এই ফেরত যাওয়ার সার্কিট চালু থাকবে। যতক্ষণ না আর একটা কার্ড দিয়ে আমরা এই লুপের বাইরে বেরিয়ে আসতে বলছি ইঞ্জিনকে। ধরুন মিলে হিশেব চলছে একটা সিরিজের। পরপর সমস্ত ধনাত্মক পূর্ণসংখ্যার বর্গের যোগফলের একটা সিরিজ। আমরা একটা মান ক পেতে চাইছি, যখন

$$ক = ১^২ + ২^২ + ৩^২ + \ldots$$

এবার ওই ফেরত পাঠানোটা ততক্ষণ অব্দিই চলবে যতদূর অব্দি পূর্ণসংখ্যা আমরা চাইব।

ধরুন আমরা এই সিরিজটাকে হিশেব করতে চাইলাম ১০০ অব্দি, ঠিক করলাম যে, ১০০-ই হবে এই সিরিজের সবচেয়ে বড় পূর্ণসংখ্যা। তার মানে ফেরত পাঠানোটা চলতে থাকবে যতক্ষণ-না বর্গ করছি যে পূর্ণসংখ্যাটার তার মান ১০০ হয়। এই খেয়ালটা রাখছে ওই ফেরত পাঠানোর কার্ডটাই, ১০০ নামক সংখ্যাটাকে সে রেখে দিয়েছে স্টোর-এ। যেই বর্গ করার সংখ্যাটা এই রেখে দেওয়া সংখ্যাটার সমান হচ্ছে অমনি সে লুপটাকে বন্ধ করে দিল। আর

ফেরত যেতে হবেনা, এবার মিল আবার এর পরের কার্ড অনুযায়ী পরপর হিশেব করে যেতে পারবে। এরপরেই যদি থাকে যোগফলটা ফুটিয়ে তোলার আদেশ লেখা কার্ড তাহলে সে যোগফলটাকে পাঠিয়ে দেবে প্রিন্টারে। আজকের প্রতিটি কম্পিউটার ভাষার একদম প্রাথমিকতম একটা অস্ত্র এই লুপ আদতে আডার আবিষ্কার।

এখানে তো আর রাখতে পারছি না, কিন্তু অত্যন্ত উচ্চস্তরের ডকুমেন্টেশন আছে, একদম আডার লেখা মূল আদেশগুলো তুলে দিয়ে দিয়ে, জন ওয়াকারের সাইটে, অটোক্যাড যাদের তৈরি তার একজন (http://www.fourmilab.to/babbage/cards.html)। আডা লাভলেস দিয়ে সার্চ মারলে আরো বহু কিছু পেয়ে যাবেন, লেসটা কিন্তু এল এ সি ই (Lovelace), যদিও, আডার জীবনটা যা, এল ই এস এস (Loveless) হলে বোধহয় বেশি মানাত। আডার জীবনী লিখেছেন ক্যাথরিন এলিয়ট, তার সাইটেও আডা সম্পর্কে বহু কিছু আছে, বিশেষত আডার আবেগজগতের তথ্য (http://pages.cpsc.ucalgary.ca/~elliotk/Ada/Bio.html)।

আডা ইচ্ছে প্রকাশ করেছিলেন, তার কবরটা যেন হয় তার বাবার কবরের ঠিক পাশেই, হয়ত মার প্রতি বিতৃষ্ণাতেই আরো। ১৮১৬-য় বায়রন ইংলন্ড থেকেই চলে যান, তখন আডার বয়েস এক বছর। আডার আট বছর বয়েসে বায়রন মারা যান, গ্রীসে। যদিও তখনো সমাপ্ত হয়নি তার স্ত্রীর সঙ্গে তার আদালতের লড়াই। ... the child of love, though born in bitterness/ And nurtured in convulsion — প্রেমের সন্তান, তবু জন্মেছিল ক্লেদে, আশ্রিত আক্ষেপের ক্রোড়ে — এটা বায়রনের চাইল্ড হ্যারল্ডস পিলগ্রিমেজ (Childe Harold's Pilgrimage) থেকে, যেখানে আডাকে নিয়ে, আডার নাম করে ছড়ানো আছে অজস্র লাইন।

বায়রন খুব কষ্ট পেয়েছেন আডার সঙ্গে দেখা না-করতে পেরে, কিন্তু লেডি বায়রনের ক্রোধ আর তিক্ততা আডা আর বায়রনের মধ্যে পাঁচিল হয়ে দাঁড়িয়েছে, আডার মৃত্যু অব্দি। মৃত্যুশায়ীন আডার একটা ছবি এঁকেছিলেন লেডি বায়রন, সেটা ক্যাথরিন এলিয়টের সাইটে আছে, এলিয়ট আমার ইমেলের উত্তর না-দেওয়ায় ছবিটা এখানে দিতে পারলাম না। ছবিটার নিচে ডানদিকের কোণে সবচেয়ে মনোযোগের বিন্দুতে লেডি বায়রনের সই, আডার চোখ বন্ধ, রুগ্ন আঙুলে বালিশের কিনারটা আঁকড়ানো, যে আঁকছে সে তার মা, গোটা ব্যাপারটাই অনেকটা স্ট্যানলে কুব্রিকের ফুল মেটাল জ্যাকেট বা বার্গম্যানের সেভেন্থ সিলের মত, কাফকার কাসলের মত — বাংলায় কোনো শব্দ নেই, বাঙালির ইতিহাসে কোনো বিশ্বযুদ্ধ নেই বলেই হয়ত, একটাই শব্দ আসতে পারে — গ্রোটেস্ক।

তখন সদ্যআবিষ্কৃত বিদ্যা সিম্বলিক লজিক বা সাঙ্কেতিক যুক্তিবিজ্ঞানের জমিতে আডার চাষবাসের সবচেয়ে বড় প্রমাণ শর্তাধীন লাফ বা কন্ডিশনাল জাম্প। কার্ড-পাঠের যন্ত্রটাকে দিয়ে আর একটা কায়দা আবিষ্কার করলেন আডা। এখানে আর নির্দিষ্ট কিছু পুনরাবৃত্ত কার্ডের সমাহারকে তুলে রাখা এবং পরে যখন যখন তাদের প্রয়োজন তখন নামিয়ে ব্যবহার করা নয়, এবার আদেশ দেওয়া হবে কার্ড রিডার যাতে লাফ দিয়ে চলে যায় যে কোনো কার্ড সমাহারের যে কোনো জায়গায়, যদি — খেয়াল করুন — শব্দটা স্ক্রিনে আঙুল রেখে পড়ুন — যদি — একটা নির্দিষ্ট শর্ত বা কন্ডিশন পালিত হয়। এই এক'টা 'যদি' প্রোগ্রামিং চিন্তনের মানে কম্পিউটার চিন্তনের জগতে সবচেয়ে বড় ভাঙচুরগুলোর একটা। কন্ডিশনাল জাম্প।

নিন্দুকেরা বলে, কম্পিউটার জগতে তার অবদান অনেকটা রান্নাবান্নার জগতে রাঁধার পরে খাবার তুলে রাখবার বাটির মত, মানবসভ্যতার ইতিহাসে উর্বরতম বেনেদের একজন, বিল গেটস, সেই তিনিও একবার, এমনকি, একটা হাই লেভেল কম্পিউটার ভাষা বানানোর চেষ্টা করেছিলেন, কম্পিউটারের ইতিহাসে খাজাতম ভাষার একটা, সেই বেসিকে একটা ৬৪ এমবি জেনিথে নিউমেরিকাল অ্যানালিসিসের নিউটন-র‍্যাফসন মেথডের প্রোগ্রাম লেখার চেষ্টা করেছিলাম, হয়নি, বারবার ইনফিনিট লুপে পড়ে যাচ্ছিল, ঠিক জীবনের অবশিষ্ট জায়গায় আমার অভিজ্ঞতার মতই, তাতে আমরা এই লাফটা দিতাম 'গো-টু' (`goto`) বলে একটা স্টেটমেন্ট দিয়ে। ধরুন এইরকম — ইফ বোর হচ্ছেন গোটু পরের দিন। এই যাঃ, এটা আমি সত্যি সত্যি বলেছি নাকি।

এই ছোট্টটা 'যদি'-টা যোগ হওয়া মাত্র ইঞ্জিনের কাজের এলাকা বিশুদ্ধ পাটীগণিত ছেড়ে বেরিয়ে এল, শুধু ওই রাশি রাশি যোগ বিয়োগ গুণ ভাগ গুষ্টির পিণ্ডিই করে যাওয়া নয়, এখন সে সিদ্ধান্ত নেয়, যত কুটীরশিল্প আকারেই হোক, তৃতীয় বিশ্ব রকমেই হোক, ইঞ্জিনটা এখন, অর্থপূর্ণ সম্ভাব্যতা নিয়েই, ডিসিশন নেওয়ার ক্ষমতার অধিকারী।

আডা কিন্তু লিখেছিলেন, কোনো একদিন হয়ত ভিক্টোরিয়ান যুগের প্রকৌশলের থেকে অনেক অনেক বেশি মাত্রার সামর্থ নিয়ে মেশিন তৈরি হবে, এবং সেদিন এই অর্থপূর্ণ সম্ভাবনাগুলো কী চূড়ান্ত আকার নেবে তা নিয়ে জল্পনাও করেছেন আডা — কোনোদিন কি সত্যিকারের বুদ্ধিমান মেশিনও বানানো যাবে? আর্টিফিশিয়াল ইন্টেলিজেন্স বা যন্ত্র-মনন নিয়ে তার মন্তব্যকে প্রায় একশো বছর বাদে তুলে এনেছিলেন অ্যালান টুরিং, আর একটা উন্মাদ প্রতিভা, আডার বক্তব্যকে টুরিং ডেকেছেন লেডি লাভলেসের আপত্তি বলে। আডার যুক্তি ছিল এই যে, "অ্যানালিটিকাল ইঞ্জিন এই ভান করেনা যে সে কিছু সৃষ্টি করতে পারে। সে শুধু তাই-ই করতে পারে যা আমরা জানি কী ভাবে তাকে করতে বলতে হবে।"

হয়তো অ্যানালিটিকাল ইঞ্জিনের অসমাপ্তির হতাশা এখানে একটা ইন্ধন আকারে কাজ করে থাকতে পারে, আডা ক্রমে চূড়ান্ত ভাবে জুয়ার নেশায় আক্রান্ত হয়ে পড়েন। গোটা জীবনে একমাত্র যার কাছ থেকে কিছুটা যত্ন পেয়েছেন সেই লর্ড লাভলেসের প্রত্যক্ষেই প্রথম দিকে, পরে ক্রমে তাকে গোপন করে। ব্যাবেজও ক্রমে জড়িয়ে পড়েন আডার এই জুয়ার চক্করে। আডার এই অ্যানালিটিকাল ইঞ্জিনে উৎসাহের একটা জায়গা ছিল এই দিয়ে ঘোড়দৌড়ের নম্বর হিশেব করতে পারা। দু-দুবার লর্ড লাভলেসের পারিবারিক গয়না গোপনে বন্ধক দেন আডা, ধার জমে যাওয়া বুকিদের ব্লাকমেল বন্ধ করার চেষ্টায়। আডার জীবনের চূড়ান্ততম হেরে যাওয়ার মুহূর্তও আসে এই সময়েই — বন্ধক দেওয়া গয়না ফেরত পাওয়ার জন্যে যখন তাকে নিজের মা লেডি বায়রনের কাছে হাত পাততে হল, লর্ড লাভলেসকে গোপন করে।

এই জুয়ার কাজে ডিফারেন্স ইঞ্জিনকে ব্যবহারও করেছেন ব্যাবেজ আর আডা। এই করতে করতে একদিন, মর্গে কি হৃদয় জুড়োলো, আডা ক্যান্সারে পড়লেন। ব্যাবেজ তার পরেও অনেকদিন বেঁচে ছিলেন। অ্যানালিটিকাল ইঞ্জিন আর তিনিও শেষ করতে পারলেন না, হয়তো আডার অনুপস্থিতিও একটা প্রাণশক্তি নিয়ে নিয়েছিল তার। অসুস্থতায় আর মৃত্যুতে উজ্জীবিত আডার এই জীবনের সঠিক কাহিনী লিখতে গেলে আডা-ব্যাবেজের বন্ধু, মানুষের এবং ঈশ্বরের আভ্যন্তরীন শুভতায় বিশ্বাসী, চার্লস ডিকেন্সকে দিয়ে চলত না, লাগত এডগার অ্যালেন পোর মত কোনো তান্ত্রিক কাপালিক। আডা মৃত্যুর আগে ব্যাবেজকে লিখেছিলেন, অ্যান্ড প্রে ডু নট কোয়াইট ফরগেট মি, ফর আই থিংক দেয়ার সিমস সাম ডেঞ্জার অফ ইট। রক্ত ক্লেদ বসা থেকে রৌদ্রে ফের উড়ে যায় মাছি — তবুও তো প্যাঁচা জাগে

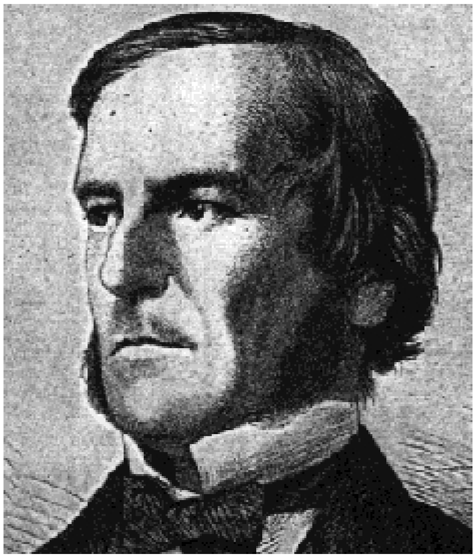
২.২।। জর্জ বুলি — চিন্তার ব্যাকরণ

কমলা জিগেশ করেছিল হেরমান হেসের সিদ্ধার্থকে, যুবক, কী পারো তুমি? কী-টাকে এখানে নাগরী লীলাসুন্দরী লাস্যময়ীর ঠোঁট বেয়ে গড়িয়ে পড়া কৌতুহলের ঔদ্ধত্যের অ্যাকসেন্ট সহ পড়তে হবে। সিদ্ধার্থ বলেছিল, আই ক্যান থিংক, আই ক্যান ওয়েট, আই ক্যান ফাস্ট। উনবিংশ শতাব্দীর ব্রিটেনের কলোনিয়াল অ্যাকাডেমিয়া ঠিক এই প্রশ্ন করেছিল জর্জ বুলিকে। বুলি কোনো উত্তর দিতে পারেননি, গরিব ঘরের ছেলে, অত সাহিত্য পড়বেন কখন? আর আনফরচুনেটলি, উপন্যাসটা তখনো লেখা বা লেখার পর জর্মন থেকে অনুবাদ হয়নি, হেরমানের তখনো জন্মই হয়নি।

বুলি বা বুলির লজিক সাইবার ইতিহাসের সিনে ল্যান্ড করেছিল বড্ড দেরিতে, অ্যানালিটিকাল ইঞ্জিনের ডিজাইনের পিছনে সমাসীন কম্পিউটার ভাবনাটাকে বদলানো আর তখন সম্ভব না। কিন্তু পরের যুগের পরের শতাব্দীর সবচেয়ে নাটকীয় ঘটনা কম্পিউটারের ঘটে ওঠার পিছনে আডা-ব্যাবেজের ঠিক পরেই যার নাম আসে তিনি জর্জ বুলি (১৮১৬-১৮৬৪)। বুলি ছিলেন আডা-ব্যাবেজের সম্ভ্রান্ত আভিজাত্যের জগত থেকে অনেকটা দূরে।

১৮৩২-এর এক বিকেলে ঘাসজঙ্গলের মধ্যে দিয়ে নড়তে নড়তে, আকস্মিক, কেকুলে যেমন হঠাৎ একটা স্বপ্নে, একটা সাপের নিজের লেজ গেলার চেষ্টা দেখে, খুঁজে পেয়েছিলেন জৈবযৌগের কার্বন বন্ডের রহস্য, সতেরোর সদ্যযুবক জর্জ বুলির মাথায় গজিয়ে উঠেছিল সেই তত্ত্ব যার গোটাটা নিজেই বুঝে বুঝিয়ে এবং লিখে উঠতে তাকে সময় নিতে হবে আরো বাইশ বছর — 'অ্যান ইনভেস্টিগেশন অফ দি ল-জ অফ থট' বেরোবে ১৮৫৪-য়। এই গজিয়ে ওঠাটা বুলির নিজের কাছেও এতটা অব্যাখ্যনীয় যে এর কথা বলতে গিয়ে তিনি 'আনকনশাস' শব্দটা ব্যবহার করেন। বুলির এই গজিয়ে ওঠা আইডিয়াকে ক্রমে একটা পূর্ণাঙ্গ বিদ্যার রূপ দেন বুলি নিজেই। বুলির লজিকের

আলোচনা করতে গিয়ে বার্ট্রান্ড রাসেল লিখেছিলেন, বিশুদ্ধগণিত মানে পিওর ম্যাথামেটিক্স আসলে উদ্ভাবন-ই করেছেন বুলি।



জর্জ বুলি

ওই প্রথম যৌবনেই বুলি লক্ষ করেন, মানুষের যুক্তিনির্মাণ বা রিজনিং বা লজিক-কে অনেকটাই ধরা যায় বীজগণিত বা অ্যালজেব্রা দিয়ে। এবং তার এই বীজগাণিতিক সমীকরণ দিয়ে চমৎকার সমাধান করা যায় লজিকের যে কোনো সমস্যা। কিন্তু অন্য একটা সমস্যা সমাধানের চাবি বুলির হাতে একটুও ছিলনা। তোমার পিতৃপরিচয় না থাকলে তুমি যাই বলো তোমার কথা কেউ শুনবে না। পিতৃপরিচয় মানে পরিচয় দেওয়ার মত একটা পিতৃকুল। তোমার বাপ গরিব মুদি — এটা স্ট্যান্ডার্ড একটা পরিচয় হল? বুলি যাই বলুন, যতই বিধবংসী এবং বিপ্লবী, তাকে পাত্তা দেওয়ার মত কাউকে পাওয়া গেলনা ব্রিটিশ অ্যাকাডেমিতে। আর একটা সমস্যা ছিল এই যে লজিককে তখনো একটা গাণিতিক এলাকা বলে চিনতেই শেখেনি গণিতজ্ঞরা। সেটাই তারা শিখল, বুলির কাছে, গত দেড়শো বছর ধরে। এমনকি বুলি যখন ছাপালেন তার প্রথম লেখা সেটাও কেউ পড়ল না, তার মৃত্যুর অনেক পর অব্দিও বুলিকে পড়তে শেখেনি পৃথিবী। ১৮৪৮-এ 'কেমব্রিজ অ্যান্ড ডাবলিন ম্যাথামেটিকাল জার্নাল'-এ বেরোনো 'দি ক্যালকুলাস অফ লজিক' অন্তত গোটা পঁচিশেক ওয়েব সাইটে এখন সগর্বে প্রদত্ত, এর একটা, এইচটিএমএল পিডিফ লাটেক ডিভিআই সব ফর্মাটেই দেওয়া আছে — http://www.maths.tcd.ie/pub/HistMath/ — গিয়ে দেখুন, আধুনিক সভ্যতার ইতিহাসের একটা জ্যান্ত খণ্ড, কত গল্প রয়ে গেছে লেখাটার সঙ্গে। যদি না-ও পড়েন লেখাটা, ব্রাউজারে খুলে স্ক্রিনে একটু আঙুল বোলাবেন। দেলুজ আর গুয়াতারি লিখেছিল না, সভ্যতা এগিয়েছে বেদনা থেকে বেদনাতরতায়, সঙ্কট থেকে সঙ্কটতরতায়?

পৃথিবী বুলিপাঠ শিখল অনেক পরে, যখন কম্পিউটার ক্রমশ জন্মে উঠছে, সময়ের ভাঁজে ভাঁজে লুকিয়ে থাকা সমস্ত টুকরো টুকরো উপাদানগুলো একত্রে মিলে যাচ্ছে, একটা তখনো-অদৃশ্য ছকে, কদিন বাদেই দৃশ্য হয়ে উঠবে, কম্পিউটারের কর্মমান জীবন্ত শরীরে। ছকের অনুপস্থিত জায়গাগুলো মেলাতে গিয়ে পাগলের মত অনুসন্ধান করে চলেছেন সব, চার নম্বর দিনে যে বিজ্ঞানীদের সঙ্গে আমরা পরিচিত হব, ভন নয়মান, আইকেন, জুসে, মশলে — এরা সব, এবং এদেরও আগে, অ্যালান টুরিং। কম্পিউটার বানানোর একদম কাঁচা বাস্তবিক কাজটা করতে গিয়ে ইলেকট্রিকাল ইঞ্জিনিয়াররা দেখলেন, তাদের এমন এক ধরনের গাণিতিক যন্ত্রপাতি দরকার যা আপাতত নেই।

পদার্থবিদ্যা এবং গণিতের মধ্যে এই ধরনের পারস্পরিকতা বারংবার ঘটেছে। একদম সেই ওয়েভ পার্টিকল ডুয়ালিটির সময় থেকে। এর সবচেয়ে নাটকীয় উদাহরণ নন-ইউক্লিডিয় জ্যামিতি। কোথায় কোন পাগলের মত সব স্বতঃসিদ্ধের বিদঘুটে এক জ্যামিতি খাড়া করলেন, করেই রাখলেন, নিকোলাই ইভানোভিচ লোবাচেভস্কি (১৭৯২-১৮৫৬), গেয়র্গ ফ্রিডরিখ বার্নহার্ড রিমান (১৮২৬-১৮৬৬) — এরা সব। আনতাবড়ি সব পাগলামি — কারোর কোনো কাজে লাগেনা। আর একজন, ছোটবেলা থেকেই যার মাস্টারমশাইরা বলে আসছেন, ছেলেটার বুদ্ধিটা তেমন

কিছু ইয়ে নয়, পৃথিবীকে চিন্তাকে দর্শনকে একমুহূর্তে বদলে দেওয়ার মত একটা তত্ত্ব বানালেন, গত শতাব্দীর শুরুতে, জিটিআর। জেনারেল থিওরি অফ রিলেটিভিটিটা থিওরি আকারে ঘটেই উঠতে পারত না ওই নন-ইউক্লিডিয়ান জ্যামিতিগুলো না থাকলে। কারণ, দেখা গেল, আইনস্টাইনের তত্ত্ব মোতাবেক ব্রহ্মাণ্ডের মডেলটা দাঁড় করানো যায় একমাত্তর ওই জ্যামিতি দিয়েই।

বিদ্যুৎ ইঞ্জিনিয়ারিং-এর প্রয়োজনীয় গণিতটা এবার দিল বুলির বীজগণিত, তার সময় তার পৃথিবী যাতে কান দেয়নি। নতুন ধরনের এই কম্পিউটার তৈরি করতে গিয়ে, ১৯৩০-এর দশকে, ইলেকট্রিকাল ইঞ্জিনিয়ারদের তৈরি করতে হচ্ছিল সার্কিটের একটা জটিল জাল, যার প্রত্যেকটা সার্কিট-খণ্ড এক একটা সুইচ এই অর্থে যে, কোনো না কোনো একটা বিদ্যুৎ প্রবাহকে কোনো না কোনো ভাবে নিয়ন্ত্রণ করে ওই খণ্ডটা, প্রত্যেকটা খণ্ডই। মোট সার্কিটের এই সবগুলো খণ্ডকে, সুইচকে, মিলিয়ে তৈরি হয় কম্পিউটারের বৈদ্যুতিক স্নায়ুতন্ত্র, যা দিয়ে বিদ্যুৎ-প্যাকেট যায় কম্পিউটারের মধ্যেই এক ঠাঁই থেকে আর এক ঠাঁই — বিদ্যুৎ সিগনাল দিয়ে তথ্য। এবার ওই বিদ্যুৎ মানে তথ্যকে যদি ওই সুইচদের দিয়ে নিয়ন্ত্রণ করাতে হয় তাহলে দ্বিধাহীন রকমের নিখুঁত ভাবে ওই সার্কিটাংশগুলোর ব্যবহারকে হাজির করতে হবে বীজগাণিতিক সমীকরণ দিয়ে। ব্যবহার বলতে — কী নেয় আর কী দেয় এক একটা সুইচ — কতটা বিদ্যুৎ কী ভাবে ঠিক কোথা দিয়ে ঢোকে তাদের ভিতর, এবং কতটা কোথায় কী ভাবে বেরিয়ে আসে। কারণ ওই বিদ্যুৎপ্যাকেটগুলোই তথ্য। তথ্য মানে তথ্য এবং আদেশ। বিদ্যুৎ স্পন্দন দিয়ে এখন থেকে বিধৃত এবং বিবৃত হবে 'এবং' (AND), 'অথবা' (OR), এবং সবচেয়ে গুরুত্বপূর্ণ ওই 'যদি' (if)। এর সঙ্গে ক্যালকুলেটরসই 'যোগ' 'বিয়োগ' 'গুণ' 'ভাগ' তো আছেই। এবার তাহলে এমন একটা নতুন ধরনের বীজগাণিতিক সমীকরণ লাগবে যাদের দিয়ে কম্পিউটার সার্কিটের এই যাবতীয় ব্যবহার এবং ধর্মকে হাজির করা যায়।

এখানে বৈদ্যুতিক সার্কিটগুলো একদম যুক্তিবিদ্যা বা লজিকের প্রক্রিয়াতেই কাজ করছে, তাই লজিকের সমস্যা মেটানো যায় এমন কোনো যন্ত্রপাতি যদি পাওয়া যায় — তাদের দিয়েই এই বৈদ্যুতিক সার্কিটের কাজও করা যাবে, সার্কিট এবং লজিক এই দুইয়েরই অপারেটরগুলো যখন কাজ করছে একই নিয়মে। ১৯৩০-এর দশকের সেই শেষের দিকে, তখন কেউই জানেনা এমন কোনো গণিতের কথা যা দিয়ে লজিকের এইসব কাজকর্ম করা যায়। সময়ের ভাঁজে গোপন রয়ে যাওয়া ছকগুলো খুঁজে পেতে একটা সঠিক পরিপ্রেক্ষিত এবং সঠিক মন লাগে। তখন ম্যাসাচুসেটস ইনস্টিটিউট অফ টেকনোলজির ওস্তাদ ছাত্র, পরবর্তীকালে ইনফরমেশন থিওরির পথিকৃৎ, ক্লদ শ্যানন, খুঁজে পেলেন বুলির বীজগণিত — বৈদ্যুতিক সার্কিটের কাজকর্মের সঙ্গে খাপে খাপে মিলে যাওয়া একটা আস্ত গাণিতিক তত্ত্ব। তৈরি হল ডিজিটাল কম্পিউটারের বিসমিল্লা।

আজীবন দারিদ্রতাড়িত, নিজে নিজে শিক্ষিত হয়ে ওঠা একটা মাস্টারের, আডার সঙ্গে একই বছরে জন্ম, অথচ যার জুয়া তো দূরের কথা, বইপত্র কেনার পয়সাও থাকত না সবসময়, জনবিস্মৃত কালবিস্মৃত ওই একপিস তত্ত্ব যদি না-থাকত, লজিক আর গণিত আর সার্কিটের ভিতর ওই যোগটা তৈরি হতে পারতনা। প্রায়-গজিয়ে-ওঠা কম্পিউটার পেটে নিয়ে প্রতীক্ষা করতে হত পোয়াতি ধরিত্রীকে, যদ্দিন না কোনো বুলি আসে, মানে বুলির কাজটা করে দেয় আর কোনো বুলি। বুলির অস্ত্রে বলীয়ান হয়ে আধুনিক ইলেকট্রনিক কম্পিউটার যেরকম স্টেরয়েডী দৌড় শুরু করল সেটা তখন সম্ভবই হতনা। অথচ সেই বেচারা মাস্টার কিন্তু কোনোদিন একটা ভ্যাকুয়াম টিউব বা ইন্টিগ্রেটেড সার্কিট চোখেও দেখেনি — দেখতে গেলে তাকে একটু বড্ড বেশি দীর্ঘজীবি হতে হত।

ডিরাক তার ম্যাগনেটিক মনোপোল বা চৌম্বক একমেরুর জন্যে যতটা জনচিত্তে সেনসেশনাল, তার চেয়েও বেশি বোধহয় এই গল্পে যে, ল্যাবরেটরিতে ডিরাক ঢুকলেই, তত্ত্বের তাত্ত্বিকতার তাপ ডিরাকের চারপাশে ছড়িয়ে পড়ত এতটাই, ইনস্ট্রুমেন্টগুলো সব ভুলভাল রিডিং দিতে শুরু করত। ডিরাক মানেই — এনিথিং থিওরেটিকাল চলছে, এনিথিং প্রাক্টিকাল চলবেনা। মেশিনের সঙ্গে ডিরাকের এতটাই আশনাই ছিল বলে শোনা যায়। বুলির বেলায় এরকমটা যদি নাও ঘটে, অন্তত ব্যাবেজ ছিলেন ইঞ্জিনিয়ার, আর বুলি ছিলেন গণিতজ্ঞ। উঁচুনিচু ঘাসজমিতে হাঁটতে হাঁটতে নৃমুণ্ডের আবছায়ায় ওইসব গজিয়ে ওঠা, আর তারপর সেগুলোকে তত্ত্বে নামিয়ে আনার জন্যে নিজেকে দুই দশক ধরে প্রয়োজনীয় গণিতটা শিখিয়ে তোলা, এরপর যে অ্যাকাডেমিয়াবিস্মৃত তত্ত্বটা তৈরি হয়েছিল, আজ সেটাই হয়ে দাঁড়াল সফটওয়ারের অভ্যন্তরে চিন্তাপ্রক্রিয়ার বিমূর্ত যৌক্তিক গঠন আর ইলেকট্রনিক যন্ত্রের মূর্ত ভৌত বাস্তবতা — এইদুটোর ভিতর একমাত্র সংযোগ।

ব্যাবেজ আর বুলি এই দুজন কাজে এসেছেন দুটো আলাদা স্বপ্ন নিয়ে, কিন্তু সেই দুটো স্বপ্নেরই একত্র মিলিত আকার আপনার টেবিলের এই পিসিটা যার স্ক্রিনে এটা আপনি পড়ছেন। তার নিজের সামাজিক জাগতিক অবস্থান, বা অবস্থানের অভাব, যাই হোক, বুলির স্বপ্নের অভাব ছিলনা। বুলির এক ছাত্র অনেক পরে তার স্মৃতিচারণে লিখেছিলেন, উনি পড়াতেন যখন, মনে হতনা একজন শিক্ষক ব্ল্যাকবোর্ডে অঙ্ক কষাচ্ছে, মনে হত একজন শিল্পী ক্যানভাসে ছবি আঁকছে তুলি দিয়ে। এখানে বুলির যে ছবিটা আমরা দিলাম সেটাই সর্বত্র পাওয়া যায়, অনেকটা সেই হাতিবাগানে জুতো কেনা শ্যামবাজারের শ-সম্পন্ন ভূপর্যটক মন্দার বোসের রোলে কামু মুখার্জির মত দেখতে লাগছে, কিন্তু তাও, হয়তো ভাবতে চাইছি বলেই ভেবে নিচ্ছি, চোখদুটো আলাদা।

মাত্র ষোল বছর বয়েসে পরিবারের অর্থনৈতিক অনটন বুলিকে বাধ্য করেছিল ইশকুলের চাকরি নিতে, ছেলেদের অঙ্ক শেখাতে শেখাতেই নিজেরও শেখার শুরু। আরও অঙ্ক তো তাই যা শেখার জন্যে পয়সা লাগেনা, ল্যাবরেটরি নেই, অল্প দু-একটা বই হলেই হয়। সতেরোয় বুলির কাছে উদ্ভাসিত হয় গণিত ও বিমূর্ত চিন্তার সেই নতুনতর তত্ত্ব। বুলি তার কুড়ি বছর বয়েসে এসে খুঁজে পান সেই জায়গাটা যা তার সময়ের সবচেয়ে বড় গণিতজ্ঞরাও ধরতে পারেননি — অপরিবর্তনীয়তা বা ইনভ্যারিয়েন্সের একটা বীজগাণিতিক তত্ত্ব, যা, পরে, অনেক পরে, আইনস্টাইনের রিলেটিভিটি তত্ত্বের একটা স্তম্ভ হয়ে দাঁড়ায়।

১৮৪৯ অব্দি বুলি পড়িয়েছেন ওই এলিমেন্টারি স্কুলেই, তারপর তার ছাপা হওয়া লেখাগুলোর ভিত্তিতে কাজ পান আয়ারল্যান্ডের কর্কে, কুইনস কলেজে। তারও পাঁচ বছর পরে বেরোয় ‘অ্যান ইনভেস্টিগেশন অফ দি ল-জ অফ থট’, যার ভিত্তিতে লেখা ‘ম্যাথামেটিকাল থিওরিজ অফ লজিক অ্যান্ড প্রোবাবিলিটি’। ইউক্লিড যেমন জ্যামিতির একদম গোড়ার বক্তব্য এবং নিয়মগুলোকে তৈরি করে দিয়েছিলেন তার স্বতঃসিদ্ধ আর উপপাদ্য, অ্যাক্সিওম আর থিওরেম দিয়ে, ঠিক তেমনি জর্জ বুলি বীজগাণিতিক চিহ্ন দিয়ে যুক্তিবিজ্ঞানকে বোঝার একদম গোড়ার জায়গাটাকে তৈরি করে দিয়েছিলেন। বুলি বিশ্বাস করতেন যে মানুষের যৌক্তিকতার ভিত্তিটাই হল লজিক।

গণিত আর বৈদ্যুতিক ইঞ্জিনিয়ারিং এই দু দুটো বিষয়কে আমূল বদলে দেওয়ার পরেও, বুলির এই তত্ত্বের সবচেয়ে জরুরি জায়গাটা কিন্তু দাঁড়িয়ে আছে বেশ সহজ কিছু নীতির উপর। ঠিক যে অ্যালজেব্রা আমরা ইশকুলে শিখি সেটা থেকেই শুরু করেছেন বুলি, তারপর অ্যালজেব্রার সাধারন নীতিগুলোয় সামান্য কিছু তাৎপর্যপূর্ণ বদল — এই দিয়েই তিনি লজিকের কলকব্জাগুলোকে এনে ফেলেছেন বীজগণিতে।

যুক্তিবিজ্ঞান বা লজিক, আর হিশেব বা ক্যালকুলেশন, এই দুটো আপাত অনন্বিত জগতকে একত্রে এইরকম ঘেঁটে দেওয়ার জন্যে যে গাণিতিক তত্ত্ব ব্যবহার করেছেন বুলি সেই জগতে রাশি মাত্র দুটো — ব্রহ্মাণ্ড বা ইউনিভার্স, আর শূন্যতা বা নাথিং। তাদের জন্যে চিহ্ন দুটো হল ১ আর ০। বুলি তখনো জানতেন না, আপনি জানেন এখন, বারবার ইতিমধ্যেই আমাদের কথা হয়েছে, এই তত্ত্বটাই তৈরি করে দিল একটা দ্বিরবস্থা-কাঠামো, টু-স্টেট-সিস্টেম, যা দিয়ে একদিকে যেমন লজিককে নিয়ে আসা যায় পরিমাপযোগ্য সংখ্যার জগতে, তেমনি ভ্যাকুয়াম টিউব বা রিলে-সিস্টেম ইত্যাদি ভৌত উপাদানের ক্রিয়াকেও চমৎকার চিত্রন করা যায়, যাদের কাজটা সবসময়েই ঘটে এইরকম দ্বিরস্থায়, একটা অবস্থা যখন বিদ্যুৎ আছে মানে হ্যাঁ মানে ১, অন্য অবস্থাটা হল যখন বিদ্যুৎ নেই মানে না মানে ০।

বুলির বীজগণিতের এই চিহ্ন আর ক্রিয়া, সিম্বল আর অপারেশন দিয়ে লজিকের সমস্যাকে অনাবিল এনে ফেলা যায় বীজগণিতের সমীকরণে, এবং তারপর তাদের সমাধান করে দেওয়া যায় ধ্যাড়াধ্যাড় করে, জাস্ট কিছু সিম্পল অ্যালজেব্রা দিয়েই। সেই অপারেশন বা ক্রিয়াগুলো একেবারেই গণিতের, কিন্তু তা দিয়েই তখন সমাধান করা যায় লজিকের সেই সমস্যাগুলোকে যা আসলে চিন্তন প্রক্রিয়া সঞ্জাত।

মানুষের যুক্তিবোধ চিন্তা আর লজিক এতটাই সম্পৃক্ত বলে, বুলি তার এই তত্ত্বকে ডেকেছিলেন যুক্তিবোধের গণিত বা ‘ক্যালকুলাস অফ রিজনিং’ বলে। মানুষের যুক্তিবোধকেই যেন তিনি ধরে ফেলছেন তার গণিত দিয়ে। বুলির ধাত্রীবিদ্যায় যে ছানা ভূমিষ্ঠ হয়েছিল, আজ সে রীতিমত সোমত্থ, আজ আমরা জানি, মানুষের যুক্তি বা রিজন নামক যন্ত্রটা এই বীজগণিতের কাঠামোয় কষে ফেলার তুলনায় অনেক বেশি ক্যাঁচালপ্রবণ, জটিল, দ্বর্থশীল, নিয়মভাঙা, এবং শক্তিশালী — কনফ্লিকচুয়াল, কমপ্লিকেটেড, অ্যামবিগুয়াস, আনপ্রেডিক্টেবল, এবং পাওয়ারফুল। আর আরো একটা জটিল খেলা চলে যুক্তি শব্দার্থ ছবি এবং ধারণাকে মিলিয়ে — শব্দের অর্থ বা ধারণা যেখানে তার প্রদত্ত অর্থের

অন্টলজিকাল খাঁচা থেকে মানে শব্দ থেকে নিরন্তরই খসে যাচ্ছে, ধারণা যেখানে তার আধার মানে শব্দ থেকে শুধুই সরে যাচ্ছে, স্থগিত থাকছে, ডেফার ও ডিফার করছে, জাক দেরিদা যাকে ডেকেছেন দিফেরঁস বলে। সেই জটিল খেলাটা এবার এই যুক্তিবোধের আত্মসচেতনতার প্রশ্ন নিয়ে আসছে, যুক্তিবোধ দিয়ে যুক্তিবোধকে বোঝার প্রাথমিক প্যারাডক্সে ঠেলে দিচ্ছে গোটাটাকে। যাই হোক, সেইসব ক্যাঁচালপ্রবণ জায়গা নিয়ে কথা বলা জন্যের এই জিএলটি ইশকুল পাঠমালা নয়, আমরা ফেরত আসি বুলির গণিত এবং কম্পিউটারের কথায়।

উনবিংশ শতাব্দীর প্রকৌশলের জগতের পক্ষে আডা-ব্যাবেজ বা বুলি কাউকেই মননগত আশ্রয় দেওয়া সম্ভব ছিলনা। তাদের কাজের উপ্ত সম্ভাব্যতাকে তার ঘটমানতায় ফুটিয়ে তোলার পক্ষে টেকনোলজিটা ছিল বড্ড বেশি দুধেভাতে। আজকের ইলেকট্রনিক ভাবে নির্মিত ডিজিটাল কম্পিউটারে যেটা সম্ভব হয়েছে। আডা-ব্যাবেজ এবং বুলি সময়ের ভাঁজে তাদের অসমাপ্ত এবং আপাত অসফল কাজগুলোকে রেখে যাচ্ছিলেন — পরবর্তী সময়ের যোগ্য পাঠকের জন্যে — ক্যাচ মি ইফ ইউ ক্যান। স্পিলবার্গের ক্যাচ-মি-ইফ-ইউ-ক্যানের ফ্র্যাংক যেমন, বুক ভেঙে যাওয়া বেদনা ও প্রতিভার সমাহার নিয়ে এক সতেরোর কিশোর, আডা-ব্যাবেজ আর বুলিও তাই। হয়ত তাই, জিএলটির পাঠমালার জন্যে ততটা প্রয়োজনীয় না-হওয়া সত্ত্বেও আমি না-লিখে পারলাম না। আওয়ার সুইটেস্ট সংস আর দোজ দ্যাট টেল অফ স্যাডেস্ট থট। এই দিনের আলোচনার একটা বড় অংশই আমি আপনাদের জন্যে লিখছিলাম না, লিখছিলাম নিজেরি জন্যে, না-লিখে পারছিলাম না, সত্যিই।

২.৩।। বুলির লজিক আর শ্যাননের সার্কিট

একটা জিনিষ বাকি আছে এখনো — পরের যুগের ইনফরমেশন থিওরির প্রবক্তা ক্লদ শ্যানন (১৯১৬-২০০১) যেভাবে বুলিকে আনলেন, কম্পিউটারের সার্কিটের প্রয়োজনীয় গণিতের আকারে সেটা নিয়ে দু-একটা কথা বলার আছে। এখানে না বললেও চলত, কারন নেটে প্রচুর পাওয়া যায় এই বিষয়ে। কিন্তু আমাদের মধ্যমগ্রাম জিএলটির, গ্নু-লিনাক্স ঠেকের প্রথম মিটিং-এ গৃহীত প্রস্তাবে যেমন আমরা বলেছিলাম, নেট তো তৃতীয় বিশ্বে থেকেও নেই। যেখানে উপায় আছে নেটে পৌঁছনোর সেখানেও নেই, কারণ, নেটে সাইট যাদের, যারা ওয়েব পেজ লেখে, তাদের জীবনযাপনটা ঘটে অন্য ভাষায়, দূরত্বটা শুধু বাংলা আর ইংরিজির নয়, দূরত্বটা সংস্কৃতির, ইয়ার্কির, পিছনে লাগার, মূল্য বিচারের, সবকিছুর। তাই সেই অর্থেও একটু দায়িত্ব থাকে বৈকি, অন্তত একটু ইঙ্গিত দিয়ে রাখার।

এর পরেও যদি আপনাদের কারো আর একটু জানতে ইচ্ছে করে, যোগাযোগ করতে পারেন, বুঝি-না-বুঝি, অনেক কিছুই নামিয়ে রেখেছি বা যোগাড় করে রেখেছি, যা নেটে পাওয়া যায়। কিন্তু, দেখুন, আমি বেজায় আস্তিক, ক্রমহ্রাসমান ত্রাসে মেনে থাকি সরকার আইন এবং ভগবানকে, তাই কোনো কিছুই আপনাকে সিডিতে লিখে দিতে পারব না, জীবনে কখনো আইন ভাঙিনি কিনা, কী করব। তবে আমার বাড়িতে বসে যদি পড়তে চান, আমার মেশিনে, কোনো আপত্তি নেই। নিজের বিড়ি নিজে আনবেন, প্লিজ। এই ক্লদ শ্যানন আর বুলির চক্কর নিয়ে বেজায় ভালো একটা পেজ আছে, http://www.cs.ucf.edu/courses/cgs3269/Lecture/notes/Chapter/203-1.pdf। আরো অনেক আছে, এটা পিডিএফে। সুন্দর করে ছবি দিয়ে বোঝানো। তবে দু-একটা ছাপার ভুলও আছে, এই লেখাটার সঙ্গে মিলিয়ে পড়ুন।

আমরা বিজ্ঞানপ্রতিভা এবং চ্যাংড়ামির সমান্তরাল চূড়ান্ততার উদাহরণ হিশেবে ফেইনম্যানের কথা জানি। কম্পিউটারের জগতেও তার কমতি নেই। ভন নয়ম্যান বিখ্যাত ছিলেন তার উপশমহীন 'এ' জোকস আর অশ্লীল ছড়ার উদার ভাণ্ডারের জন্যে। ক্লদ শ্যাননের একটা গল্প খুব চলে, বেল ল্যাবরেটরিতে থাকাকালীন তার দুহাতে বল জাগলিং করতে করতে সাইকেল চালানোর। দাবার গেমস, মেজ বা গোলকধাঁধার গেমস, লোকঠকানো মাইন্ড রিডিং গেমস বানানোয় শ্যানন অনেক সময় ব্যয় করেছেন। ক্রিপ্টোগ্রাফি বা সঙ্কেতবিদ্যাকে একটা সিরিয়াস জায়গা দিয়েছিল তার ১৯৪৯-এর প্রবন্ধ 'কমিউনিস্ট থিওরি অফ সিক্রেসি সিস্টেমস'। আসা যাক বুলির গণিতকে কী ভাবে শ্যানন আনলেন আধুনিক ডিজিটাল কম্পিউটারের জগতে সেই কথায়।

আজকের জগতের সবচেয়ে গতিশীল বিজ্ঞানগুলোর একটা হল ইনফর্মেশন থিয়োরি বা তথ্য-বিজ্ঞান। এই তথ্যবিজ্ঞানের গোড়াপত্তন শ্যাননের হাতে। যদি কেউ চান, শ্যাননের '৪৮ সালের সেই লেখাটা পড়ে দেখতে পারেন, নেটে দেওয়া আছে, ফ্রি ডাউনলোড করা যায়, http://cm.bell-labs.com/cm/ms/what/shannonday/paper.html

থেকে, ‘এ ম্যাথামেটিকাল থিওরি অফ কমিউনিকেশনস’। শ্যাননের তথ্য-বিজ্ঞান তথ্য বলতে বোঝে শুধু সেইসব চিহ্নের সমাহারকে যাদের মধ্যে অস্তিত্বগত ভাবেই একটা অনিশ্চয়তা আছে। অনিশ্চয়তা এই অর্থে যে সেই চিহ্নটা নিশ্চিত ভাবে কী হবে তা আগে থেকে বলা সম্ভব নয়। ধরুন, আমাদের টেলিগ্রামগুলোকেই, সেখানে ‘এ’ বা ‘দি’ এই ইংরিজি আর্টিকলগুলোকে অনেক সময়ই আমরা ইচ্ছেমত বাদ দিয়ে দি, কারণ, খুব সহজেই, নিশ্চিতভাবেই, বুঝে নেওয়া যায়, এখানে কী আসবে। সেই একই ভাবে, একটা বাক্যের মোট শব্দের মোট বর্ণমালা থেকেও অনেক অনাবশ্যক বর্ণকে উড়িয়ে দেওয়া যায়, তারা নিশ্চিত বলে, এবং তাই অপ্রয়োজনীয় বলে। ‘শধ দরকর কথ লখ রখ’ — এই কথাটায় ‘শুধু দরকারি কথা লিখে রাখো’ এই অর্থ বোঝার কোনো অসুবিধে বোধহয় হয়না, স্বরবর্ণগুলো তুলে দেওয়া হয়েছে বলে। তবে এতে একটা অসুবিধে হতে পারে, ধরুন ‘ক’ আর ‘থ’ দিয়ে তৈরি মোট শব্দের সংখ্যা যদি খুব বেশি হয়, এবং সম্ভাব্য শব্দগুলোর মধ্যে ‘কথা’ ছাড়াও অন্য শব্দ চলে আসে, তখন আর এভাবে মানে বোঝা যাবেনা। অর্থাৎ, তখন আর ওই স্বরবর্ণগুলো অপ্রয়োজনীয় বা নিশ্চিত থাকবেনা।

শ্যাননই আমাদের প্রথম জানিয়েছিলেন যে অনিশ্চয়তাই হল সেই জিনিষ যা আমরা ভাবের আদানপ্রদান বলে বুঝে থাকি। বেশ জবরদস্ত লাগছে না? জীবনের সবচেয়ে গভীর আদানপ্রদানগুলোর সঙ্গে খুব সুগভীর মিল পাচ্ছেন না? অবশ্য সবার অভিজ্ঞতা তো একরকম হয়না। যাইহোক, এখানে আর এসব কপচানো যাবেনা। আমরা কুপথে চলে যাচ্ছি, বুলির গণিত থেকে শ্যাননের ইলেকট্রনিক্সের মূল আলোচনা ছেড়ে।

ক্লদ শ্যানন

বুলির তত্ত্ব মোতাবেক, আমরা কোনো একটা সীমাবদ্ধ এলাকার ভিতর থেকে যখন কোনো নির্বাচন করি তখন সমস্ত সম্ভাব্য পছন্দগুলোকে ব্যবহার করি। ধরুন, একটা পাত্রে সাদা আর কালো দুরকম বল আছে, দুই রঙের বলই আছে দুই সাইজের, ছোট আর বড়। এবার আমাদের বলা হল, ‘বড় কালো’ বল তুলতে। ভাষাগত ভাবে এই ‘বড় কালো’ শব্দবন্ধে ‘বড়’ এবং ‘কালো’ এই দুটো নিরিখ রয়েছে। এদের দুই ভাবে নেওয়া যেতে পারে — ‘বড় এবং কালো’, আর ‘বড় বা কালো’। বুলির কথা অনুযায়ী আমরা ধরে নেব — ‘বড় এবং কালো’। ‘বড় AND কালো’।

বিপরীতে, যদি আমাদের বলা হত, ‘সব বড় কালো বলকে বাদ দাও’, আমরা সেখানেও একাধিকভাবে সমস্যাটাকে দেখতে পারতাম, ‘কালো এবং বড়’ বল বাদ দাও, বা ‘কালো বা বড়’ বল বাদ দাও। কিন্তু এক্ষেত্রে আমরা যে নিরিখটা খুঁজে নেব তা হল ‘কালো কিন্তু বড় নয়’ — ‘কালো NOT বড়’।

এই দুটো ক্ষেত্রেই আমরা সমস্ত সাদা বলকে বাদ দেব কারণ তারা আমাদের সঠিক নিরিখ নয়। যদি আমরা চাইতাম সমস্ত কালো বল এবং সমস্ত ছোট সাদা বল, তাহলে আমাদের নিরিখটা হত — ‘(কালো AND বড়) OR (সাদা NOT বড়)’। আমরা যখন ইন্টারনেটে গিয়ে সার্চ মারি, সেটা বুলিয়ান খোঁজা, সেই খোঁজাখুঁজিতেও এইরকম লজিকই ব্যবহার করা হয়। যদি আমরা ‘জর্জ AND বুলি’ খুঁজি তাহলে ব্রাউজারটা এমন সব পাতা খুঁজবে যেখানে ‘জর্জ’ আর ‘বুলি’ দুটো শব্দই আছে। আর যদি আমরা ‘জর্জ OR বুলি’ খুঁজতাম, তাহলে এমন সমস্ত পাতা খোঁজা হত যেখানে হয় ‘জর্জ’ বা ‘বুলি’ এই দুটো শব্দের যে কোনো একটা আছে, মানে হয় ‘জর্জ’ বা ‘বুলি’ বা ‘জর্জ’ এবং ‘বুলি’ দুটোই।

এবং আমাদের খোঁজার নিরিখটা কী হচ্ছে — খেয়াল করুন — থাকা আর না-থাকা, বাইনারি একটা দ্বিরবস্থা। সত্যি/মিথ্যে, হ্যাঁ/না, ১/০, অন/অফ, খোলা/বন্ধ ইত্যাদি। এর পরে আরো এক রকম লজিক গজিয়েছে — ফাজি

লজিক, কিন্তু ঈশ্বর আমায় ত্রান করুন, আপনারা যা পড়ছেন, এটা লিখতে গিয়ে আমাকে তার বিশগুণ পড়তে হচ্ছে এটা বুঝতে পারছেন? তাতেও তো লাগ-এর তথাগত সঙ্কর্ষণরা নিয়তই ভুল পাচ্ছে — ভাল্লাগে না। এই ফাজি লজিক বা ঘোলাটে যুক্তি আংশিক সত্যি তাই আংশিক মিথ্যেকে খুঁজে নিতে পারে।

বুলির এই লজিক সাঙ্কেতিক বা সিম্বলিক এই অর্থে ধরুন ক বা খ ইত্যাদি চিহ্ন দিয়ে বাস্তব জীবনের কোন বাস্তব বস্তুটাকে বোঝাচ্ছি তাতে কিছু মাথা না-ঘামিয়ে শুধু ক বা খ হিশেবে তাদের নাড়াচাড়া করেই তাদের যুক্তিগত সমস্যার সমাধান করা যায়। এই ক বা খ নামক সঙ্কেতই তার কাজ করার জন্যে যথেষ্ট। ধরুন একটু আগে যখন ক দিয়ে আমরা একটা পলিনোমিয়ালের মান বার করছিলাম, তখন সেটাকে আমরা ডাকতে পারতাম চলরাশি বা ভ্যারিয়েবল বলে। বুলির গণিতে তারা ভ্যারিয়েবল নয়, সিম্বল। এবার ওই সিম্বলগুলোকে নাড়াচাড়া করার, তাদের উপর বিভিন্ন ক্রিয়ার বা অপারেশন করতে পারার বা না-পারার কিছু নিয়ম আছে — সেই নিয়মগুলোকে নিয়েই বুলির গণিত। এই চিহ্নগুলোকে কোনো একটা নির্দিষ্ট মূর্ত বাস্তবতার সাপেক্ষে বোঝানো বা ইন্টারপ্রিট করা যেতেই পারে, যেমন কালো বল বা সাদা বল দিয়ে, এদেরকে এবার কোনো একটা চিহ্ন দিয়ে বোঝানো যেতেই পারে।

কিন্তু এর সঙ্গে চালু বীজগণিতের চলরাশি বা ভ্যারিয়েবলকে গুলিয়ে ফেলবেন না, এটা কিন্তু এমন একটা গণিত যেখানে ১ + ১ = ১ হয়। অন্যান্য বীজগণিতের চলরাশিদের কাছে যা স্যাক্রিলেজ, ভগবানকে গাল দেওয়ার মত পাপ। বুলির এই গণিতে ঠিক আমাদের পরিচিত গণিতের মতই কিছু অপারেশন বা ক্রিয়া আছে, সেই ক্রিয়াগুলো যারা করে তারা অপারেটর। তিনটে প্রাথমিক অপারেটরকে আমরা এইমাত্র আলোচনা করে এলাম — ‘AND’, ‘OR’ এবং ‘NOT’।

‘AND’ অপারেটর

বুলির গণিতে ‘AND’ অপারেটরকে বোঝানো হয় একটা বিন্দু ‘.’ দিয়ে, কিন্তু প্রায়শই এই বিন্দুটা আর লেখার সময় লাগানো হয়না। যা লেখার কথা ছিল ‘ক.খ’, তাকে লেখা হয় ‘কখ’। ক আর খ এই দুটো চিহ্ন বা সিম্বলকে কাজে লাগিয়ে চারটে আলাদা আলাদা সমাহারের সম্ভাবনাকে লেখা যায়।

(১) ক আর খ দুজনেই সত্যি

(২) ক সত্যি কিন্তু খ মিথ্যে

(৩) ক মিথ্যে কিন্তু খ সত্যি

(৪) ক আর খ দুজনেই মিথ্যে

আমাদের কালো আর সাদা বলের উদাহরণ কাজে লাগিয়ে বলা যায়, ধরুন ‘ক’ বলতে আমরা বোঝালাম, ইন্টারপ্রিট করলাম, কালো হওয়ার ধর্মটা, অর্থাৎ ‘ক’ মানে কালো। আর ‘খ’ মানে বড়। এবার ‘ক.খ’ বা ‘কখ’ মানে বোঝাবে, এই ক্ষেত্রে — ((কালো হওয়ার ধর্ম) AND (বড় হওয়ার ধর্ম))।

এবার, বুলির গণিত মোতাবেক, যদি দুটো অপারেন্ড বা কর্মের মানই ‘সত্যি’ হয়, সেই ক্ষেত্রে সর্বমোট মানটাও ‘সত্যি’ হবে। ধরুন আমরা এবার ‘বড় সাদা’ বল তুলতে চাইছি, চিহ্নের নিরিখে ক্রিয়াটা হল ক.খ। ক.খ এই ক্রিয়ার সর্বমোট মান হবে —

(১) ক সত্যি (কালো) এবং খ সত্যি (বড়) হলে ক.খ সত্যি (বড় কালো বল)

(২) ক সত্যি (কালো) এবং খ মিথ্যে (ছোট) হলে ক.খ মিথ্যে (ছোট কালো বল)

(৩) ক মিথ্যে (সাদা) এবং খ সত্যি (বড়) হলে ক.খ মিথ্যে (বড় সাদা বল)

(৪) ক মিথ্যে (সাদা) এবং খ মিথ্যে (ছোট) হলে ক.খ মিথ্যে (ছোট সাদা বল)

এর মধ্যে দেখুন ‘ক.খ’ সত্যি একমাত্র যখন ‘ক’ সত্যি এবং ‘খ’ সত্যি। কিন্তু যদি ‘ক’ মিথ্যে হয়, না-কালো মানে সাদা বলের ক্ষেত্রে, কিম্বা যদি ‘খ’ মিথ্যে হয়, না-বড় মানে ছোট বল, কিম্বা যদি ‘ক’ আর ‘খ’ দুটোই মিথ্যে হয়, না-কালো না-বড় মানে সাদা ছোট বল — এরকম সমস্ত ক্ষেত্রেই ক.খ মিথ্যে হবে। আর একভাবে বললে, ‘ক.খ’ সত্যি বা এর মান ১ হবে তখনই যখন ‘ক’ এবং ‘খ’ দুটোই সত্যি, দুটোরই মান ১।

আর একটু সহজ করে ব্যাপারটাকে লেখা যায় একটা ট্রুথ-টেবল বা সত্যি-টেবিল ব্যবহার করলে। এই সত্যি-টেবিলে দেখুন, আমরা প্রত্যেকটা সারিতে ‘ক’ আর ‘খ’ এই দুইয়ের আলাদা আলাদা মানের জন্যে ‘ক.খ’-র মান দেখিয়েছি।

‘AND’ অপারেটরের সত্যি-টেবিল		
ক	খ	ক.খ
১	১	১
১	০	০
০	১	০
০	০	০

‘OR’ অপারেটর

বুলিয়ান গণিতে ‘OR’ অপারেটরকে লেখা হয় ‘+’ চিহ্ন দিয়ে। এটাকে পাটীগণিতের অভ্যস্ত ‘+’ চিহ্নের সঙ্গে গুলিয়ে ফেলবেন না। আমাদের এই সাদা কালো বড় ছোট বলের উদাহরণটা বড্ড বেশি ক্লাসকনসাস হয়ে গেছে, এবং একটু রূপকথাও, বল থাকে বলবানের, আর বলবান তো সাদা হয়, বড় লোক। তাও, এটাকেই ব্যবহার করা যাক, একবার যখন করে ফেলেছি। ‘ক + খ’ বলতে এবার বোঝাবে ওই বলের পাত্র থেকে সেইসব বলকে তুলে নেওয়া যারা কালো অথবা বড় অথবা কালো এবং বড় দুইই। বুলির গণিত অনুযায়ী, যদি কোনো একটা অপারেন্ড সত্যি হয় বা দুটোই সত্যি হয়, তাহলে সর্বমোট অপারেশনটা সত্যি হবে। এবং, এর মানে, আমরা তুলে নেব সমস্ত কালো বল, তাদের আকার নির্বিশেষে, বড় আর ছোট দুই-ই, আর বড় সাদা বলগুলোকে। পাত্রে অবশিষ্ট থাকবে শুধু ছোট সাদা বলগুলো। এবং এই গণিতে দেখুন, ১ + ১ = ১। আমরা এর কথায় আসছি। একটু বাদেই।

‘OR’ অপারেটরের সত্যি-টেবিল		
ক	খ	ক + খ
১	১	১
১	০	১
০	১	১
০	০	০

‘NOT’ অপারেটর

আগের দুটো অপারেটর ‘AND’ এবং ‘OR’ এই দুটো অপারেশনেই অপারেন্ড-এর সংখ্যা দুই। কিন্তু ‘NOT’ অপারেটর কাজ করে একটিমাত্র অপারেন্ড-এর উপর। যে অপারেন্ডকে নাকচ করে বা উল্টে দেয় এই ‘NOT’ অপারেটর। বা, আর এক ভাবে বললে, সত্যি অপারেন্ডকে মিথ্যে করে, বা মিথ্যেকে সত্যি করে। ‘ক’-এর ‘NOT’ অপারেটর লিখি আমরা ‘ক '’ চিহ্ন দিয়ে। বা, ‘ক’-এর উপরে একটা দাগ দিয়ে,‘ক̄’। এখানে আমরা ‘NOT’ অপারেটর-এর সত্যি-টেবিল দেখালাম।

‘NOT’ অপারেটরের সত্যি-টেবিল	
ক	ক '
১	০
০	১

বুলির মৃত্যুর প্রায় আশি বছর বাদে, ক্লদ শ্যানন, শ্যাননের নিজের বিনয়ী ভাষায়, সেই মুহূর্তে 'লজিক আর ইলেকট্রিকাল ইঞ্জিনিয়ারিং দুটোর সঙ্গেই পরিচয় আছে' এমন এক মাত্র লোক, একে প্রয়োগ করলেন সুইচ সার্কিটের নির্মাণে। শ্যানন তার এই বুলিকে প্রয়োগের কাজ শুরু করেন একটা অ্যানালগ কম্পিউটারে। পরে যা ডিজিটাল কম্পিউটারে লাগানো হয়। অ্যানালগ বলতে কী বোঝায় তা আমরা বলেছি আগে। এই শব্দটা দিয়ে এক ধরনের হিশেব যন্ত্রকে বোঝানোর একটা ইতিহাস আছে।

এখানে অ্যানালগ এই বিশেষণটা বোঝাচ্ছে — একটা সার্কিটের বা ডিভাইসের আউটপুট আর ইনপুট পরস্পর সমানুপাতী। অ্যানালগ শব্দটা এবং আমাদের পরিচিত অ্যানালজি শব্দটা, তুল্যমূল্য তুলনা, এই শব্দ দুটো এসেছে একই প্রকৃতির উপর বিভিন্ন প্রত্যয়ে। অ্যানালগ কম্পিউটারের পিছনে চিন্তাটা এই যে বিচ্ছিন্ন একক আলাদা আলাদা সংখ্যা, যা দিয়ে আমরা এখন ডিজিটাল ক্যালকুলেটর বা কম্পিউটারে অভ্যস্ত — সেই সংখ্যাদের দিয়ে কাজ না-করে, হিশেবটা করছি যে ব্যাপারটাকে বোঝার জন্যে, সেই ভৌত ব্যাপারটারই একটা ছোট্ট ভৌত মডেল বানালাম, প্রতিটি উপাদানের একটা নির্দিষ্ট অনুপাতকে বজায় রেখে। এবার সেই ছোট মডেলটা একটা কম্পিউটার হয়ে কাজ করবে, অ্যানালগ কম্পিউটার, যেখানে একটা ভৌত ইনপুট থেকে ভৌত আউটপুট পাব, শুধু সেটাকে ততগুণ বাড়িয়ে নিতে হবে বাস্তবটাকে পাওয়ার জন্যে, বাস্তব থেকে যতগুণ কমিয়ে আমরা মডেলের ইনপুটটা দিয়েছিলাম।

অনেক অনেক জায়গায় এটা অপরিসীম রকমের জটিল হয়ে পড়বে, ধরুন আপনাকে হিশেব করতে হচ্ছে জনসংখ্যার বদলের হার, আপনার লিলিপুট পুতুলদের দিয়ে জনসংখ্যা বাড়ানোর কাজটা আপনি কতটা করাতে পারবেন আমার সে ব্যাপারে যথেষ্ট সন্দেহ আছে। কিন্তু সেটা যদি বিদ্যুৎ বন্টন ব্যবস্থার গ্রিড হয়? নিখুঁত ভাবে সেটাকে বোঝার দরকার পড়ে কারুর, পড়তেই তো পারে? সবাই তো ডব্লুবিএসইবি নয়, যেখানে হিশেব একটাই, বাংলা হিশেব, মফস্বল মরে মরুক, শহরে বিদ্যুৎ দাও, নইলে খবরের কাগজ গালাবে, আর মফস্বলে কী হচ্ছে তাতো কেউই জানেনা। ক্রাইস্ট তো শহরে এসে থেমে গিয়েছেন, ক্রাইস্ট স্টপড অ্যাট ক্যালকাটা, দুটোই 'সি' দিয়ে — এটা যাতে বোঝা না যায়, আমরা আসলে 'সি'-তে প্রিন্স চার্লস আর ক্রাইস্টেরই বাচ্চাকাচ্চা, সেই জন্যেই তো 'কে' দিয়ে করা হল কোলকাতাকে। দেখুন, আবার বাজে বকছি। কিন্তু, যা বলছিলাম, এই ধরনের হিশেবের জন্যে তৈরি ভৌত মডেল, অ্যানালগ কম্পিউটার সত্যিই হত, যেখান থেকে 'অ্যানালগ' এই ধারণাটাই এসেছে। শ্যানন এরকম মেশিনেই কাজ করেছিলেন।

ধরুন ওই বিশালাকার বিদ্যুৎ বন্টন ব্যবস্থা। তার বিদ্যুৎ প্রবাহ হিশেব করতে হচ্ছে। বেজায় জটিল হিশেব। এখন তো আমরা ডিজিটাল ক্যালকুলেটর বা কম্পিউটার দিয়ে হিশেব করি। এবার, একটা সিচুয়েশন ভাবুন যেখানে সেসব নেই, তাই, একটা মডেল বানানো হল মোট বিদ্যুৎ ব্যবস্থার, অনেক অনেক ছোট আকারে, শুধু অনুপাতটা এক রেখে। অনুপাত মানে সব কিছুরই অনুপাত, গোটা বিদ্যুৎ গ্রিডটার বিভিন্ন জায়গায় বিভিন্ন উপাদানের রোধগত প্রবাহগত বিভবগত, রেজিস্টিভ ক্যাপাসিটিভ ইনডাক্টিভ পরিমাণগুলোর সঙ্গে অনুপাত বজায় রেখে বানানো এই কুচো মডেলটাই এবার গোটা বিদ্যুৎ গ্রিডটার বাস্তব বিদ্যুৎ ব্যবস্থার একটা অ্যানালগ বা তুল্যমূল্য প্রতিরূপ হয়ে দাঁড়াল।

অনেকটা ছোট মাত্রায় এই মডেলটাই এখন একটা গ্রিড। এবার গ্রিডের যে কোনো একটা ঘটনাকে বোঝানো যাবে এই মডেলটার একটা ছোট মাত্রার ঘটনা দিয়ে। এর প্রত্যেকটা ইনপুট এবং আউটপুট এবার ওই অনুপাতটা বজায় রাখবে। সমানুপাতী হবে পরস্পরের। ধরুন মডেলের ইনপুটটা যদি মূল গ্রিডের এক লক্ষ ভাগের এক ভাগ হয়, আউটপুটটাকেও এক লক্ষ দিয়ে গুণ করে নিতে হবে।

শ্যাননের আমলে ডিজিটাল হিশেবের মেশিন না থাকায় এরকম মেশিন সত্যিই বানানো হত, বিকট বিকট সব অবিশ্বাস্য জটিলতা নিয়েই তৈরি হত মেশিনগুলো। যেমন ১৯২০ নাগাদ ম্যাসাচুসেটস ইনস্টিটিউট অফ টেকনোলজিতে বিদ্যুৎ গ্রিডের প্রবাহ মাপার জন্যে তৈরি এসি নেটওয়ার্ক ক্যালকুলেটরটার জন্যে জায়গা লাগত একটা আস্ত ঘর। এই মডেলগুলোই এক একটা অ্যানালগ কম্পিউটার, হিশেব করছে, কম্পিউট করছে বিদ্যুৎ ব্যবস্থার খুঁটিনাটি। ভৌত রকমে তার অঙ্কগুলো কষে দিচ্ছে, আমরা ভৌত ফলাফলটাকে আবার একক দিয়ে সংখ্যার মাপে মেপে নিচ্ছি।

শ্যানন যে অ্যানালগ মেশিনটায় কাজ শুরু করেছিলেন সেটা এর চেয়েও এক কাঠি উপরে, তার নাম ডিফারেনশিয়াল অ্যানালাইজার। অন্য অ্যানালগ কম্পিউটারগুলো যেখানে একটা একক কাজের জন্যেই তৈরি, ডিফারেনশিয়াল অ্যানালাইজার সেখানে একগুচ্ছ বৈজ্ঞানিক এবং ইঞ্জিনিয়ারিং সমস্যার সমাধান করতে পারত, সেভাবেই বানানো — সেই সমস্ত সমস্যা যাদের ডিফারেনশিয়াল ইকুয়েশন দিয়ে হাজির করা যায়। ডিফারেনশিয়াল ইকুয়েশন কী সেটা যারা জানেন না, ছেড়ে দিন, পরের ব্যাপারটা বোঝার জন্যে এটা কাজে লাগবে না।

এই মেশিনে হিশেবগুলো হত ডেসিমাল বা দশমিক পদ্ধতিতে, বাইনারিতে নয়। আমরা শূন্য নম্বর দিনে ডেসিমাল বাইনারি নিয়ে কথা বলে এসেছি — ডেসিমাল প্রক্রিয়া যেখানে ০ থেকে ৯ এই দশটা ডিজিট বা অঙ্ক দিয়ে সংখ্যাদের বোঝে এবং লেখে, বাইনারি বোঝে ০ আর ১ এই দুটো অঙ্ক দিয়ে। ডেসিমাল ০ মানে বাইনারি ০, ডেসিমাল ১ মানে বাইনারি ১, কিন্তু ডেসিমাল ২ মানে বাইনারি ১০, কারণ, বাইনারিতে আর ডিজিট নেই। এইভাবে ডেসিমাল ৩ মানে বাইনারি ১১, ডেসিমাল ৪ মানে বাইনারি ১০০, ডেসিমাল ৫ মানে বাইনারি ১০১ চলতেই থাকে। কিন্তু মজাটা খেয়াল করুন, মাত্র দুটো জিনিষ লাগছে আপনার বাইনারি প্রক্রিয়ায়। ০ আর ১। বিদ্যুৎ থাকা এবং না-থাকা। বা, একটা বিশেষ ভোল্টের বিদ্যুৎ মানে ০ এবং আর একটা বিশেষ ভোল্টের বিদ্যুৎ মানে ১। ধরুন, আজকের একটা বিশেষ ডিজিটাল কম্পিউটারে তিন ভোল্ট মানে ১ আর আধ ভোল্ট মানে ০। অর্থাৎ অবস্থা মাত্র দুটো। এবার, ওই ডিজিটাল কম্পিউটারের অভ্যন্তরে প্রতিটি সার্কিটাংশেই ৩ ভোল্ট ঢোকা বা বেরোনো মানে ১ ঢোকা বা বেরোনো, এবং ০.৫ ভোল্ট ঢোকা বা বেরোনো মানে ০ ঢোকা বা বেরোনো। গোটা বাস-ব্যবস্থা বা কম্পিউটারের ভিতরের তথ্য চলাচল হাইওয়েতেও ওই একই বন্দোবস্ত কায়েম থাকবে।

কিন্তু শ্যাননের ওই ঘটোৎকচে এসব ছিলনা। সে কাজ করত ডেসিমালে। এবং স্লাইড রুল বা ডায়ালের ঘড়ি যেমন পরিমাপ একক দিয়ে কাজ করে, ইঞ্চি বা ডিগ্রিতে, সেইরকম এই মেশিনে সংখ্যাদের হাজির করা হত নড়া এবং দূরত্ব, মুভমেন্ট এবং ডিসটান্স, দিয়ে। একটা ডান্ডা কতটা নড়ছে তাই দিয়ে বিভিন্ন ভ্যারিয়েবল বা চলরাশিদের হাজির করা হত। গিয়ার দিয়ে গুণ আর ভাগ করা হত। আর বিভিন্ন গিয়ারের পরিমাপের পার্থক্য দিয়ে যোগ আর বিয়োগ করা হত। আঠারোখানা আলাদা আলাদা ভ্যারিয়েবল নিয়ে কাজ করতে পারত এই মেশিন। আর ইন্টিগ্রেশন করা হত একটা গোল এবং ঘোরানো যায় এমন টেবিলের উপর পরিবর্তনীয় ব্যাসার্দ্ধে ঘুরতে থাকা একটা তীক্ষ্ণ খাঁজকাটা ঘূর্ণ্যমান চাকতি দিয়ে। যারা ক্যালকুলাস জানেন না, এই ইন্টিগ্রেশন ব্যাপারটাও বুঝলেন না, ছেড়ে দিন, কোনো ক্ষতি নেই।

চার্লস ব্যাবেজ এর প্রায় এক শতাব্দী আগে তার ডিফারেন্স ইঞ্জিনে শক্তি যোগানোর কথা ভেবেছিলেন স্টিম ইঞ্জিন দিয়ে, আর ডিফারেনশিয়াল অ্যানালাইজারে শক্তি যোগাত বিদ্যুৎ। ওই ডান্ডাগুলো নাড়ানো আর গিয়ার আর চাকাগুলো নড়ানো — এটাই কাজ ছিল বিদ্যুতের। হিশেবের কাজ করত মেশিনের ভৌত শরীরটা।

প্রতিটি নতুন হিশেবের জন্যে ডিফারেনশিয়াল অ্যানালাইজারের গিয়ারগুলোকে নতুন করে হাতে করে ঠিক জায়গায় এনে দিতে হত। শ্যানন এই ডিফারেনশিয়াল অ্যানালাইজারের এই কাজগুলো করছিলেন আর একই সঙ্গে মেশিনের কাজ করার যুক্তিবিজ্ঞানগত কাঠামো নিয়ে ভাবনাচিন্তা করছিলেন। এই দুটো এক সঙ্গে করাটাই হল সময়ের সেই নতুন মোচড় যা পুরোনো সময়ের গোপন ভাঁজগুলোকে তার চোখে ফুটিয়ে তুলল। শ্যানন ভাবলেন, এই বিশুদ্ধ ভৌত রকমে হিশেব করার প্রক্রিয়াটা যদি বদলে ফেলা করা যায় হিশেব করার একটা বৈদ্যুতিক প্রক্রিয়ায়, ভৌত অংশগুলোকে নানা প্রকারের বৈদ্যুতিক সার্কিটে বদলে ফেলে? আর কী ভাবে বানানো বা সাজানো হবে সার্কিটগুলোকে? কেন, সঙ্গে আছে জর্জ বুলি, কোমর বেঁধেছে।

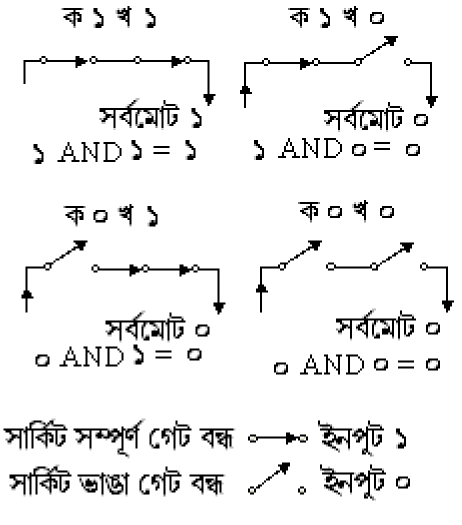
১৯৩৭-এ শ্যানন তার গবেষণা সম্পূর্ণ করেন, এবং পরের বছর এর উপর একটা পেপার লেখেন, এ সিম্বলিক অ্যানালিসিস অফ রিলে অ্যান্ড সুইচিং সার্কিটস। যাতে তিনি বুঝিয়েছেন কেমন করে লজিক সার্কিট বানাতে হয়। পেপারটা পাতে পড়ার আগে থেকেই সেনসেশন হয়ে যায় এবং দম ফেলার আগে থেকেই সেখান থেকে নতুন ধরনের ডিজাইন তৈরির প্রকৌশল বানানোর চেষ্টা শুরু হয়ে যায়। এখানে আর একটা লোকের আর তার কাজের উল্লেখ ভয়ানক দরকার, ১৯৩৬-এ বেরিয়েছিল অ্যালান টুরিং-এর সেই ডাকিনীমন্ত্রের মত শক্তিশালী বৈপ্লবিক বৈদ্যুতিক তত্ত্ব যা গোটা সাইবার চিন্তার প্রকরণটাকেই বদলে দিয়েছিল।

কিন্তু, বিশ্বাস করুন, মাইরি বলছি, আমার আর দম নেই, আজ সাতাশে নভেম্বর, ঠ্যাং প্লাস্টারের চোদ্দ নম্বর দিন। এই কাজটা শেষ করে ফেলতে হবে এই ধুমকিতে আমি রোজ কম করে ষোল ঘন্টা করে কাজ করেছি, একদিন সাড়ে উনিশ, আর তার মধ্যে এত এত নেট থেকে জিনিষ নামাতে হয়েছে যে পরের টেলিফোন বিলটা আসার পরে মানু যদি আমায় লাথি মেরে বাড়ি থেকে বার করে দেয়, প্লাস্টারে লাগানো হিলে ল্যাংচাতে ল্যাংচাতেই আমার ওকে নৈতিক সমর্থন জানাতে হবে, যেটা আরো করুণ, গালাগাল করার আরামও যখন থাকেনা।

কী ভাবে টুরিংটা লিখব সেটা ছকেও ফেলেছিলাম, ফর্মাল সিস্টেমটা বেজায় উত্তেজনার জিনিষ, একটা বিকট জান্তব বই আছে, 'গেডেল এশার বাক', এরউইন হফস্ট্যাডটের-এর লেখা। যদি কোথাও পান, চোখের পাতা না-ফেলে ঝেড়ে দেবেন, অনেকদিন পাওয়া যাচ্ছেনা। কিন্তু আমার আর এনার্জি নেই। বাড়িটা নাকি পান-বিড়ির দোকান করে ফেলেছি, 'বাবুজি' এবং বাবুজি-জাতীয় গোটা পাঁচেক গান বলির মুহূর্তে পাঁঠার ডাকের স্কেলে বেজে চলেছে, যাদের বিপিএম মানে বিট পার মিনিট একশো তো ছাই বোধহয় দুশোর উপর, যাতে বেসাল মেটাবলিক রেট নেমে না যায়, নইলে অতক্ষণ কাজ করা যায়না। আমি নয় লিখছি, কিন্তু বাড়ির অন্যদের উপরে এবার রহম খাওয়ার দরকার, তাই টুরিং ছেড়ে দিলাম, এই দিনের আলোচনাটা যদি আপনাদের ভালো লাগে মেল করবেন, পরে নাহয় করা যাবে। আর, সরাসরি এর পরের দিন কম্পিউটার প্রজন্মের আলোচনায় যেতে, বা তার পরে গ্নু-লিনাক্স কাঠামোয়, টুরিং দরকারও পড়বে না।

শ্যানন চাইলেন বুলির গণিতের ওই 'AND', 'OR' এবং 'NOT' অপারেশনগুলোকে বৈদ্যুতিক সার্কিটে নিয়ে আসতে। এইজন্যে তাকে রিলের মধ্যেকার বিদ্যুৎ সংযোগগুলোকে কিছুটা বদলে নিতে হল — এখন থেকে তারা হয় বিদ্যুৎ প্রবাহ ঘটতে দেবে — মানে সত্যি মানে হ্যাঁ মানে ১, অথবা বিদ্যুৎ-পথটা কেটে দেবে — মানে মিথ্যে মানে না মানে ০। এই বদলে নেওয়া রিলেগুলোয় এবার বিদ্যুৎ পাঠানো হবে, এতে হয় সার্কিট সম্পূর্ণ হবে, হবে বিদ্যুৎময় মানে শাট বা বন্ধ। অথবা বিদ্যুৎপথ বন্ধ থাকায় সার্কিট থাকবে বিদ্যুৎবিহীন, অসম্পূর্ণ, ওপন বা খোলা। ছবিতে দেখুন এই সময়ের রিলে সার্কিটের কানেকশনটা আমরা দেখালাম, বুলির গণিতের 'AND' অপারেটরের জন্যে।

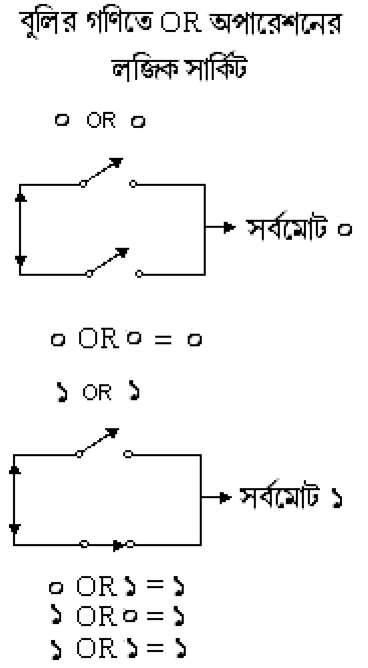
বুলির গণিতে AND অপারেশনের লজিক সার্কিট



ছবিতে দেখুন, সার্কিট যখন সম্পূর্ণ থাকছে, তখন বিদ্যুৎ যাচ্ছে এবং আমরা পাচ্ছি ১, মানে হ্যাঁ মানে সত্যি। ধরুন একটা বাল্ব যদি লাগানো থাকত এই সার্কিটে তাহলে সেটা জ্বলে উঠত। মানে ইলেকট্রনিক সার্কিট দিয়ে আমরা সেই বাইনারি গণিতকে হাজির করতে পারলাম যার উপর দাঁড়িয়ে আছে কম্পিউটারের গোটা কাজটাই। কারণ আগেই তো বলেছি, শূন্য নম্বর দিনে, তথ্য মানেই ০ বা ১, সে আঙ্কিক তথ্যই হোক, বা অনাঙ্কিক, আর তাদের উপর কাজের গোটা কাঠামোটাই খুব সরল কিছু পাটীগণিত, যাদের মিলিয়ে মিলিয়ে তৈরি হয় জটিলতর কাজগুলো।

এই ছবিতে একটা ক্রিয়াকে দেখলাম, বুলির তৈরি ‘AND’ গণিত শ্যানন কী ভাবে হাজির করলেন ইলেকট্রনিক্সে, এবার ওই একই ছবি ‘OR’ অপারেশনের জন্যে।

দুটো ছবিতেই একই ভাবে একটা উর্ধমুখী তীর দিয়ে খোলা বা কাটা সার্কিটকে বোঝানো হয়েছে, আর সোজা ডানদিকে তীর দিয়ে আস্ত সার্কিটকে, যার মধ্যে দিয়ে বিদ্যুৎ যাচ্ছে। এটা জাস্ট শুরু, বুঝতেই পারছেন, সবচেয়ে সরল সহজ প্রাথমিক আঙ্কিক ক্রিয়া, এবং ইলেকট্রনিক সার্কিটে তার প্রতিনিধিত্ব। কী করে গণিতের সমস্যাগুলোকে চালান করে দেওয়া শুরু হল ইলেকট্রনিক সার্কিটে। এবার, অন্তত, শূন্য থেকে এই তিন অব্দি লেখা তথ্য পাঠানোর ভৌত প্রকরণটা আপনি বুঝতে না-পারলেও একটা আলগা ভাসাভাসা ভাবনা তৈরি করে নিতে পারবেন।



এর একটা চমৎকার ছবি আছে আইবিএম-এর নিজের সাইটে, এখনকার একটা মাইক্রোচিপে ঠিক এই কাজটা করার জন্যে চিপের যে জায়গাটা নির্দিষ্ট তার বহু বহু গুণ বড় করে, কিন্তু কপিরাইটেড ছবি, দেওয়া যাবেনা। যাকগে, এখান থেকেই ইলেকট্রনিক কম্পিউটারের এবং আমাদের আজকের দিনের শেষ।

এবার, পরের চার নম্বর দিন থেকে আমরা ঢুকব আধুনিক ডিজিটাল কম্পিউটারের ইতিহাসে। এই অংশের রেফারেন্স অনেকগুলো সাইটের নাম আর বইয়ের নাম তো লেখার মধ্যেই দেওয়া আছে। আর গণিতের ইতিহাসের সমরেন্দ্রনাথ সেনের বিজ্ঞানের ইতিহাস একটা বিপুল বই, যদি নাও পড়েন, কিনে রাখবেন, বংশ উদ্ধার হয়ে যাবে।

আর ভারতীয় ইতিহাসের সংখ্যার প্রাচীনতম ইতিহাসের একটা খুব ভালো বই ভারি সাম্প্রতিক, আসকো পার্পোলার ডিসাইফারিং ইনডাস স্ক্রিপ্টস, সেটা মূলত সিন্ধু প্রত্নতত্ত্বের বই কিন্তু এত ধনাঢ্য যে আমি গত তিন বছর পোস্টমডার্নিজম থেকে পাঁপর অব্দি যত্রতত্র সেটা থেকে ঝেড়ে আসছি, আরো প্রচুর ঝাড়ব।

আর একটা কথা, আজকের গোটা আলোচনাটা মূলত দু জনের প্রতি কৃতজ্ঞ। বিশেষত আজকের প্রথম সেকশন এবং পরের চার নম্বর দিনের প্রায় সমস্ত ছবি নেটে এদের সাইট থেকে নেওয়া, এদের দুজনের কাছে, আরো অনেকের মত, চিঠি লিখে অনুমতি চেয়েছিলাম, এই দুজনেই উষ্ণ সম্মতি দিয়েছেন। হাওয়ার্ড রাইনগোল্ড। তার বই, দি ফুড ফর থট, নেটেই দেওয়া আছে, সাইট http://www.rheingold.com/texts/tft/1.html।

আর টনি অডসলে, তার সাইট, দি কম্পিউটার হাট, http://www.arcula.demon.co.uk/entry.htm। মানে, এই সিরিজের বিপুল সংখ্যক লেখকের মধ্যে এরাও যোগ হয়ে গেলেন।

ও, অঙ্কের ইতিহাসের সেই সাইটটা, লিংকটা পেয়ে গেছি, পুরোনো একটা ব্যাকআপে প্রথম এইচটিএমএলটা ছিল, http://www-history.mcs.st-andrews.ac.uk/history/Indexes/Hist_Topics_alph.html। গিয়ে দেখুন, মাথা খারাপ হয়ে যাবে।

সংকলন ও রচনা : মধ্যমগ্রাম জিএলটি-র (glt-mad@ilug-cal.org) তরফে ত্রিদিব সেনগুপ্ত

glt-mad@ilug-cal.org

MADHYAMGRAM

গ্নু-লিনাক্স ইশকুল

জিএলটি মধ্যমগ্রাম
glt-mad@ilug-cal.org

শূন্য নম্বর দিনে আমরা শূন্য থেকে একটা পিসিতে পৌঁছলাম। এক নম্বর দিনে আমরা সিপিইউ আর মেমরির একদম গোড়ার কিছু কথা বলে নিয়ে, একটা পিসির কাজ করে চলার মধ্যে হার্ডওয়ার, সফটওয়ার, অপারেটিং সিস্টেম ঠিক কী ভাবে সাজানো থাকে সেই নিয়ে আলোচনা করলাম। দুই নম্বর দিনে, ওই কাজ করে চলতে গিয়ে যে আইও বা ইনপুট আউটপুট ডিভাইসগুলোকে ব্যবহার করতে হয় তাদের সঙ্গে অপারেটিং সিস্টেমের দেওয়া-নেওয়া নিয়ে আলোচনা করলাম। তারপর, তিন নম্বর দিনে আমরা পিসির এই অবস্থায় পৌঁছনোর পিছনের ধারাবাহিকতাটা নিয়ে কথা বললাম — যন্ত্র এবং মনন দুই দিক দিয়ে। আমরা পিসির প্রাগ-ইতিহাস থেকে ইতিহাসে পৌঁছলাম। আজ আমরা পিসির ইতিহাস নিয়ে কথা বলব। পিসিতে আমরা পৌঁছলাম কোন বিবর্তনের মধ্যে দিয়ে, যন্ত্রের বিবর্তন, যন্ত্রকে ব্যবহার করার তরিকার পরিবর্তন, যন্ত্রের সঙ্গে যন্ত্রব্যবহারকারীর সম্পর্কের পরিবর্তন। যে অপারেটিং সিস্টেম নিয়ে দু-চারটে গোড়ার কথা আমরা এক আর দুই নম্বর দিনে বলে এসেছি, সেই অপারেটিং সিস্টেমটা গজিয়ে উঠল কী করে সেটা নিয়েই আজ কথা বলব আমরা। অপারেটিং সিস্টেম ঠিক কী এবং কী ভাবে কাজ করে, অপারেটিং সিস্টেমের শ্রেণীবিভাগ, ইত্যাদি। আমরা শুরু করব একদম আধুনিক ডিজিটাল কম্পিউটারের জন্মমুহূর্ত থেকে। প্রথমে সেই জন্মের অজস্র নাম-জানা এবং নাম-না জানা ধাত্রীদের ভিতর থেকে কয়েকজনকে নিয়ে একটু আলাদা করে কথা বলে নেব। তারপর এই প্রথম ছানা মানে জেনারেশনটা কী ভাবে পরবর্তী প্রজন্মে বেড়ে উঠতে শুরু করল তার কাহিনী। এটাও আবার এক অর্থে, আমাদের এই গ্নু-লিনাক্স ইশকুলের পাঠমালার সাপেক্ষে একটা প্রাককথন, যা শেষ হবে আজ। পরের দিন, মানে পাঁচে আমরা সরাসরি গ্নু-লিনাক্সে ল্যান্ড করে যাব। আজকের একদম শেষদিকে গিয়ে আমরা গ্নু-লিনাক্সের শুরুর শুরু সম্পর্কে কিছু কথা জানব। অবশিষ্টটা পরের দিন।

।। দিন চার ।।

১ ।। শুরুর কয়েকজন মানুষ এবং মেশিন

তিন নম্বর দিনে আমরা শ্যাননের লজিক সার্কিটের তত্ত্বের জন্মমুহূর্তে শেষ করেছিলাম। ওখান থেকে আধুনিক ডিজিটাল কম্পিউটারের প্রাগ-ইতিহাসের শেষ এবং ইতিহাসের শুরু। ইতিহাসে এর পরের স্তরেই আসেন ১৯৩০ নাগাদ হাওয়ার্ড আইকেন তার মার্ক ওয়ান নামে মেশিন নিয়ে। মেকানিকাল বা যান্ত্রিক কম্পিউটারের এপিসোড এবার শেষ, শুরু নতুন কাহিনী। প্রথমে ইলেকট্রিকাল বা বৈদ্যুতিক, পরে ইলেকট্রনিক বা বৈদ্যুতিন মেশিনের পালা। এখানেই কিন্তু শেষ নয়। কম্পিউটার বদলাতেই থেকেছে, বদলাচ্ছে আজো, কিছু বদল বাজারের নিয়ম মেনে, প্রতিযোগী কোম্পানির মেশিনের সঙ্গে আমার মেশিনের বাজার দখলের লড়াই, সেই থেকে আসা বদল, আর কিছু বদল বিজ্ঞান তথা প্রযুক্তির আভ্যন্তরীন বদলের কারণে। এই দুরকম বদল আবার জল-অচল বর্গও নয়। তারা এ অন্যের সঙ্গে জড়িত। আবার বদল শুধু যন্ত্রাংশ মানে ভৌত উপাদানের স্তরেও নয়, কম্পিউটার বদলাচ্ছে তার ভৌত গঠন বা হার্ডওয়ার, এবং কাজের আভ্যন্তরীন চিন্তাপ্রক্রিয়ার গঠন বা সফটওয়ার — এই দুই নিরিখেই। এই বদল আজো চলছে। কখনো বদলটা স্বাভাবিক, দুলকি চালে, কখনো খুব নাটকীয়।

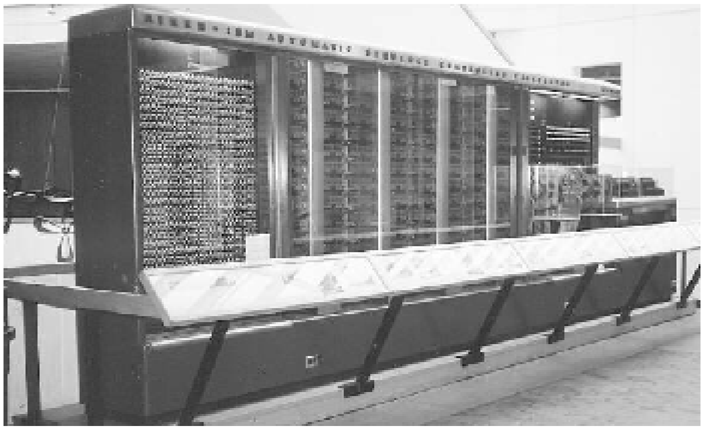
১৯৪৮ নাগাদ পদার্থবিদ ওয়াল্টার হাউজার ব্রাটেন, জন বারডিন এবং উইলিয়ম ব্রাডফোর্ড শকলে আসেন তাদের নতুন আবিষ্কার ট্রানজিস্টর নিয়ে। তার প্রকোপে আমূল এবং নাটকীয় বদলে যায় কম্পিউটারের ইতিহাস। ১৯৬০-এর দশকের শেষ দিকে আসে আইসি বা ইন্টিগ্রেটেড সার্কিট। ১৯৭০-এর দশকে আসে মাইক্রোপ্রসেসর। কিন্তু এর প্রত্যেকটা উপাদানই আমাদের আলোচনার যথাস্থানে আসবে, প্রজন্ম বা জেনারেশন থেকে জেনারেশনে কম্পিউটারের ইতিহাসের বদলের কাঠামোর সঙ্গে মিলিয়ে। সেটা নিয়ে আজ আমরা কথা বলব। প্রথমে, একদম গোড়ার কয়েকজন পথিকৃত এবং তাদের মেশিনের ছোট্ট করে উল্লেখ।

চল্লিশের দশকের মাঝামাঝি নাগাদ, মূলত যুদ্ধবাজির সরকারি তাগিদেই, প্রায় একই সঙ্গে এই ভাবনাচিন্তা শুরু হয় আমেরিকার হার্ভার্ড প্রিন্সটন পেনসিলভেনিয়া বিশ্ববিদ্যালয়ে এবং জার্মানিতে। হার্ভার্ড বিশ্ববিদ্যালয়ে হাওয়ার্ড

আইকেন, প্রিন্সটনে ইন্সটিটিউট ফর অ্যাডভান্সড স্টাডিজ-এ ভন নয়ম্যান, পেনসিলভ্যানিয়া বিশ্ববিদ্যালয়ে উইলিয়ম মশলে, এবং জার্মানির কনরাড জিউসে — এরা সবাইই এক একটা হিশেব-যন্ত্র খাড়া করেন। হাওয়ার্ড আইকেনের এই মেশিন তৈরি করে দিয়েছিল আইবিএম। হাঙ্গেরিজাত মার্কিন বৈজ্ঞানিক জন ভন নয়মানের মেশিনে প্রথম গোটা কাজটা, কাজের পদ্ধতি সহ, ভরে রাখা যেত মেমরিতে। কিন্তু সেই অর্থে প্রথম সফল কেজো কম্পিউটার যন্ত্র তৈরি হতে হতে লেগে গিয়েছিল ১৯৪৫-এ পদার্থবিদ জন মশলে আর ইঞ্জিনিয়ার জে প্রেস্পার একার্ট-এর ওই মেশিন অব্দি।

১.১।। আইকেন, হপার, মার্ক ওয়ান

মার্ক ওয়ান কম্পিটার আইকেনের নামের সঙ্গেই জড়িত, কিন্তু আইকেনের পাশাপাশি আর একজনের নাম একই ভাবে আসা উচিত, তিনি গ্রেস হপার (১৯০৬-১৯৯২)। কম্পিউটার জগতে 'বাগ' এই শব্দটা, মানে প্রোগ্রামের পোকা, অগোচরে রয়ে যাওয়া ছোট ছোট গোলযোগ, যারা প্রোগ্রাম চালানোর সময় গন্ডগোল পাকাতে থাকে, গ্রেস হপারের আমদানি। সত্যিই একটা পোকা, একটা বেচারা মথ, ঘেঁটে দিয়েছিল মার্ক ওয়ানের হার্ডওয়ার। তাই, আক্ষরিক ভাবেই, প্রথম কম্পিউটারকে প্রথমবার ডিবাগ করেন, পোকা ছাড়ান, হপার। এপিটি (APT — Automatically-Programmed-Tool) বলে একটা ভাষা তৈরি করেন হপার এবং কোবল বলে পরে বহুলব্যবহৃত একটা ভাষা চালু করেন যারা তাদের একজন এই হপার। এপিটি ছিল একদম প্রথম যুগের একটা ভাষা। সংখ্যা দিয়ে নিয়ন্ত্রণ করা যায় এমন হার্ডওয়ার উপাদানগুলোকে প্রোগ্রাম করা যেত এপিটি-তে। মার্ক ওয়ানের মূল মানুষ ছিলেন অবশ্যই হাওয়ার্ড আইকেন (১৯০০-১৯৭৩)। আইকেন ছিলেন পদার্থবিদ এবং ইলেকট্রিকাল ইঞ্জিনিয়ার। হার্ভার্ডে তার ডক্টরেট শেষ করে ১৯৩৯-এ আইকেন গবেষণাতেই রয়ে যান। গবেষণার ফান্ড দিয়েছিল আইবিএম। এর দায়িত্বে যে তিনজনের ইঞ্জিনিয়ার টিম, তাতেই ছিলেন আইকেন আর গ্রেস।



মার্ক ওয়ান

আইকেনের মাথায় মার্ক ওয়ান জাতীয় বৈদ্যুতিক-যান্ত্রিক বা ইলেকট্রোমেকানিকাল মেশিনের ধারণাটার সূত্রপাত ঘটে ১৯৩৭-এ। তার সাত বছর পর, ১৯৪৪-এ, মার্ক ওয়ান নির্মাণের গোটা কাজটা শেষ হয়। ৪৭-এ আইকেন বানান মার্ক টু, এর পরের সংস্করণ। মার্ক ওয়ান ছিল একটা হলঘর ভর্তি মেশিন, ঘটাং ঘটাং করে তাতে ধাতুর যন্ত্রপাতি নড়ার আওয়াজ হত, ৫৫ ফুট লম্বা এবং ৮ ফুট উঁচু। পাঁচ টন ওজনের এই জরদগব বানাতে লেগেছিল সাত লক্ষ ষাট হাজার আলাদা আলাদা যন্ত্রাংশ। মার্কিন নেভির কামানের গোলা ছোড়ার কাজে ব্যবহার হয়েছিল মার্ক ওয়ান, পনেরো বছর ধরে, ১৯৫৯ অব্দি। কাগজের টেপ দিয়ে তথ্য স্থানান্তরের কাজ হত। পেপার টেপ থেকে তথ্য পড়ার জন্যে ছিল চারখানা পেপার টেপ রিডার। একটা রিডার প্রোগ্রাম মানে আদেশ জাতীয় তথ্য বা ইন্সট্রাকশন ডেটা পড়ার, অন্য তিনটে হল বিশুদ্ধ কাঁচা তথ্য বা র ডেটা পড়ার, ডেটা রিডার। পেপার টেপ ছিল একদম আইবিএম পাঞ্চড কার্ডের সমান চওড়া, মানে ৩.২৫ ইঞ্চি ।

মার্ক ওয়ান যোগ বিয়োগ গুণ ভাগ করতে পারত, এবং হিশেব প্রক্রিয়ায় পুরোনো ফলাফলকে হিশেবের মধ্যে নিয়ে আসতে পারত। মানে আগের হিশেবের আঙ্কিক তথ্য ধরে রেখে পরের হিশেবের মধ্যে নিয়ে আসতে পারত।

লগারিদম আর ট্রিগোনোমেট্রির ফাংশন ব্যবহারের জন্যে আলাদা আলাদা সাবরুটিন ছিল। দশমিকের পরে ২৩-টা অবস্থান অব্দি সংখ্যা চিবোতে পারত মার্ক ওয়ান। তথ্য ভরা এবং যান্ত্রিক ভাবে হিশেব করা — মানে কম্পিউটিং-এর কাজের জন্যে ব্যবহার হত তিন হাজার চাকা, চোদ্দশো চল্লিশ খানা ডায়াল-আকৃতির গোল সুইচ, এবং গোটা মেশিনে ব্যবহৃত মোট তারের পরিমাণ ছিল পাঁচশো মাইল। ইলেকট্রিক রিলে ব্যবহৃত হয়েছিল। সমস্ত আউটপুট ফুটিয়ে তুলত একটা ইলেকট্রিক টাইপরাইটারে। একটা গুণ করতে লাগত তিন থেকে ছয় সেকেন্ড। লগ বা ট্রিগোনোমেট্রির হিশেব করতে লাগত মোটামুটি এক মিনিট বা তার বেশি। গোটা যন্ত্রের যান্ত্রিক কাজটা করত মেশিন বরাবর লম্বায় ডান্ডায় লাগানো একটা চার অশ্বশক্তির ইঞ্জিন। আজকের আপনাকে ধরুন তুলে নিয়ে ওই হলঘরে ঢুকিয়ে দেওয়া হল, এবং জিগেশ করা হল, যন্ত্রটা কী বলোতো?

১.২।। জন ভন নয়মান এবং ফলিত গণিত

জন ভন নয়মানের জন্ম ২৮ ডিসেম্বর, ১৯০৩-এ, হাঙ্গেরির বুদাপেস্তে, মৃত্যু ৮ ফেব্রুয়ারি, ১৯৫৭, ওয়াশিংটন ডিসিতে। খোদার কুদরতে নয়মানের প্রতিভার চিহ্ন ফুটে উঠতে শুরু করেছিল তার ছ-বছর বয়েসেরও আগে থেকে, যখন ধ্রুপদী গ্রীক ভাষায় বাবার সঙ্গে জোকস বিনিময় শুরু করে ইয়ানুস, মার্কিন ভাষায় জনি। এবং বাড়ির লোক স্তম্ভিত হয়ে যেত বাচ্চা জনির আস্ত আস্ত টেলিফোন ডিরেক্টরি মুখস্থ রাখার ক্ষমতায়। ১৯২৩ অব্দি বার্লিন বিশ্ববিদ্যালয়ে রসায়ন পড়ার পর নয়মান যান জুরিখ, সেখানে কেমিকাল ইঞ্জিনিয়ারের ডিপ্লোমা পান। জুরিখ থেকেই শুরু নয়মানের গণিতপ্রেমের। বুদাপেস্ত বিশ্ববিদ্যালয় থেকে গণিতে ডক্টরেট করেন ১৯২৬-এ, সেট থিয়োরির উপর তার তত্ত্বে। পোস্ট-ডক্টরাল করতে যান সেই রূপকথা হয়ে যাওয়া গোটিংগেন বিশ্ববিদ্যালয়ে, জার্মানির গোটিংগেনকে রূপকথা করে তুলেছিলেন যারা তাদের প্রমুখ ডেভিড হিলবার্টের ছাত্র হয়ে। ডেভিড হিলবার্ট (১৮৬২-১৯৪৩) সেই গণিতজ্ঞ-দার্শনিক, ১৮৯৯-এ বেরোনো যার 'দি ফাউন্ডেশনস অফ জিয়োমেট্রি' ইউক্লিডের জ্যামিতির জায়গায় একটা নতুন জ্যামিতিকে নিয়ে আসার কাজটাকে সম্পূর্ণ করেছিল, ইউক্লিডের থেকে সম্পূর্ণ স্বতন্ত্র একটা স্বতঃসিদ্ধমালায় নির্মিত একটা নতুন জ্যামিতি — যার উল্লেখ করেছিলাম তিন নম্বর দিনে।

তদ্দিনে নয়মান স্টার থেকে সুপারস্টার হওয়ার পথে। জ্যামিতি এবং টোপোলজিতে তুমুল কাজ করা অধ্যাপক অসওয়াল্ড ওয়েবলেন তাকে ডেকে নেন কোয়ান্টাম থিওরির কাজে। ১৯৩১-এ নয়মান প্রিন্সটন বিশ্ববিদ্যালয়ে অধ্যাপক হিশেবে যোগ দেন। বেধড়ক উজ্জ্বল ছাত্র না-হলে নয়মানের কাছে পড়া প্রায় অসম্ভব ছিল। নয়মানের একটা প্রিয় খেলা ছিল তার ছাত্রদের কাছে সন্ত্রাস — একটা বিতিকিচ্ছিরি রকমের জটিল সমীকরণ বোর্ডের এককোণে বিদ্যুৎবেগে সমাধান করেই, ছেলেমেয়েরা লেখার আগে সেটা মুছে ফেলা, তারপর সেই সমাধানের স্টেপগুলো নিয়ে আলোচনা। নাইটক্লাবের ক্যাবারে পার্টির সার্কিটে এমন ছেদহীন উপস্থিতি ছিল তার যে তাকে ক্যাবারে-দুনিয়ারই নাগরিক বলে উল্লেখ করা আছে নয়মান বিষয়ে এক স্মৃতিচারণে। আর আদিরসায়নে নয়মানের বিশেষজ্ঞতার কথা তো আগেই বলেছি। নয়মানের বিরাট বিরাট কাজ আছে মেজার থিওরি, স্ট্যাটিস্টিকাল মেকানিক্স, কোয়ান্টাম মেকানিক্স-এ, টোপোলজিতে, গেম থিয়োরিতে, নন-লিনিয়ার পার্শিয়াল ডিফারেনশিয়াল ইকুয়েশনে। জীবনের শেষ বছরগুলোর একটা বড় সময় কাটিয়েছিলেন সাইবারনেটিক্সে, অটোমাটার কাজে। অটোমাটা হল অটোমেশনের

বহুবচন, মানে রোবট জাতীয় যে কোনো স্বয়ংক্রিয় যন্ত্র। এদের আভ্যন্তরীন গঠনের গণিতটাই হল অটোমাটা থিওরির ভিত্তি। এখানেই কাজ করেছেন ভন নয়মান।

১৯৩০-এর দশকে হাইড্রোডিনামিক্স বা তরলগতিবিদ্যা নিয়ে কাজ শুরু করেন নয়মান। অভ্যস্ত গাণিতিক কায়দাকানুন দিয়ে এর সমীকরণগুলোকে কব্জা করা সম্ভব ছিলনা। নয়মান তখন এদের নিউমেরিকাল অ্যানালিসিস দিয়ে সাইজ করার ধরার কথা ভাবেন। নিউমেরিকাল অ্যানালিসিস বলতে, এক কথায়, অতিসরলীকৃত রকমে — গণিত দিয়ে একটা সমীকরণকে সরাসরি সমাধান করার চেষ্টা না-করে, পরপর অনেকগুলো ধাপে একটা সমীকরণের একটা অ্যাপ্রক্সিমেট বা মোটামুটি সমাধানের আঙ্কিক মানে পৌঁছনোর চেষ্টা করা। এর একটা কারণ, বহু সমীকরণ গাণিতিক রকমে সমাধান করা যায়না, বা বহুসময় অকল্পনীয় জটিলতা জড়িয়ে থাকে তার সঙ্গে। কাজের প্রয়োজনটা মিটে যাওয়াই এখানে বড় কথা, দশমিকের পরে কয়েক ঘর অব্দি সমাধানটার মান পেলেই যদি চলে যায়, তাহলেই হল। তাত্ত্বিক ভিত্তিটা এখানে এই যে ইনফিনিট বা অনন্ত-সংখ্যক ধাপের পর এই পাটীগাণিতিক মানের অ্যাপ্রক্সিমেট বা মোটামুটি সমাধানটা মূল সমাধানের সঙ্গে বেমালুম মিলে যাবে। কিন্তু সমীকরণটাকে সরাসরি সমাধান না করে এই অ্যাপ্রক্সিমেট আঙ্কিক মানের ভিত্তিতে তাকে ইন্টারপ্রিট বা ব্যাখ্যা করার এই বহুল ব্যবহৃত পদ্ধতিটার ঝঞ্ঝাট এটাই যে একটা বিপুল পরিমাণ হিশেব কষার, কষে যাওয়ার দরকার পড়ে, প্রতিটি ধাপে। নয়মানের তাই দরকার ছিল একটা কম্পিউটারের। নয়মান, স্বাভাবিকভাবেই, একটা কম্পিউটারের গঠন কী হতে পারে তাই নিয়ে ভাবনাচিন্তা শুরু করলেন। তার ফল হল যে ছকটা সেটা ভন নয়মান আর্কিটেকচার বলেই পরিচিত। আমরা সেই শূন্য থেকে যে কাজের ভিত্তিতে কম্পিউটারের ছকটা দিয়েছিলাম — সেই বিভাগটা নয়মানই করেছিলেন প্রথম — অ্যারিথমেটিক অ্যান্ড লজিকাল ইউনিট, কন্ট্রোল ইউনিট, বিরাট আকারের একটা কেন্দ্রীয় মেমরি এবং ইনপুট/আউটপুট ডিভাইস।

এবং এর সঙ্গে সবচেয়ে গুরুত্বপূর্ণ জায়গাটা দিলেন — তুলে রাখা বা স্টোরড প্রোগ্রামের ধারণা। মানে, যে আদেশগুলো কম্পিউটারে ভরে দিতে হবে, তার তথ্যের পাশাপাশি, এটা অনুযায়ীই চলবে বা কাজ করবে মেশিন। মানে যে নিয়ন্ত্রণটা এর আগে অব্দি মানবসভ্যতার ইতিহাসে কোনো যন্ত্রকে মানুষ কোনোদিন বকলমা করতে পারেনি, তার সুযোগটা এল এইবার। তিন নম্বর দিনে আমরা বলেছিলাম হিশেব কষার শারীরিক ভার ভাগ করার জায়গা থেকে এসেছিল অ্যাবাকাস। এবার, হিশেব কষার গাণিতিক কাজের নিয়ন্ত্রণের মানসিক ভারটা তুলে দেওয়ার সুযোগ এল — কম্পিউটার নামক যন্ত্রের হাতে। যে নিয়ন্ত্রণটা নিজেদের মাথায় রাখতে হত, এখন থেকে সেটা তোলা থাকবে মেশিনের মেমরিতে — এবং এটার নামই প্রোগ্রাম। প্রয়োজনীয় আদেশগুলোকে একসঙ্গে এইভাবে তুলে রাখা বা স্টোর করে রাখতে পারা মানে প্রত্যেকবার মেশিন চালানোর আগে নতুন করে মেশিনকে আদেশ ঢোকানোর প্রয়োজন হবেনা। আর তখনকার দিনে নতুন আদেশমালা মানে কী ভাবুন — বড় বড় নতুন নতুন কানেকশন তৈরি করা নতুন নতুন তার জুড়ে জুড়ে, পরে আমরা দেখব, প্লাগবোর্ড দিয়ে। আমাদের কাছে আজকে গোটা ব্যাপারটা খুবই সাদামাঠা লাগছে। তখনকার কম্পিউটারের দুনিয়ায় এটা একটা বিরাট পদক্ষেপ। দার্শনিক ভাবেও, প্রযুক্তিগত রকমেও।

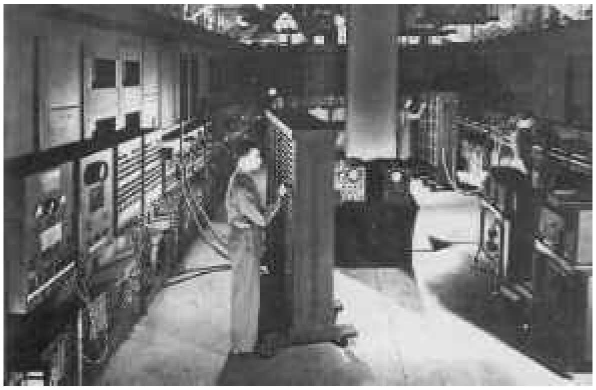
তার সময়কার অন্য অনেকের থেকে আরো অনেক বেশি করে কম্পিউটারকে একটা গাণিতিক অস্ত্র হিশেবে দেখেছিলেন ভন নয়মান। তার চিন্তনগত তাৎপর্যটা ধরতে পেরেছিলেন। বিদ্যাচর্চা তথা সভ্যতার আগামী ইতিহাস জুড়ে ফলিত গণিত বা অ্যাপ্লায়েড ম্যাথামেটিক্স-এর যাবতীয় সমস্যা সমাধানের একটা প্রবল উপায় উপায় হিশেবে কম্পিউটারের সম্ভাবনাটা আগে থেকেই ভেবে উঠতে পেরেছিলেন ভন নয়মান। হাইড্রোডিনামিক্স, ব্যালিস্টিক্স বা গোলাগুলি-মিসাইল ইত্যাদি ছোঁড়ার বিষয়ে বিজ্ঞান, মেটেরিয়োলজি বা আবহাওয়া-বিজ্ঞান, গেম থিওরি এবং রাশিশাস্ত্র বা স্ট্যাটিস্টিক্স — এই সবগুলোর উপরেই নয়মানের অগাধ এবং তুলনাহীন পাণ্ডিত্য প্রবলভাবে ব্যবহৃত হয়েছিল দ্বিতীয় বিশ্বযুদ্ধের সময়, যুদ্ধবিজ্ঞানের প্রায় প্রতিটি স্তরে। হয়তো কোনো মানে নেই, তবু এই শেষ বাক্যটা লিখতে আমার বেশ খারাপ লাগল।

নয়মান প্রথম যে কম্পিউটারের সঙ্গে সম্পৃক্ত হন সেগুলো হল মুর স্কুলের এডভ্যাক তথা এনিয়াক, যার কথায় আমরা এখুনি আসছি এবং আইকেনের ওই ম্যাক ওয়ান। ওই সময়ই নয়মান আগ্রহী হয়ে ওঠেন কম্পিউটারের রিলে সিস্টেম নিয়ে। যুদ্ধের শেষ দিক থেকে শুরু করে নয়মান বানান তার আইএএস বা ইনস্টিটিউট অফ অ্যাডভান্সড স্টাডিজ-এ তার নিজের কম্পিউটার, পরে যার মডেলে তৈরি মেশিনগুলো পরিচিত হয়েছিল 'সুপারকম্পিউটার'

বলে। পরবর্তীযুগের কম্পিউটার বিবর্তনের উপর যার বিরাট প্রভাব। প্রতিভা এবং অবদানের নিরিখে কম্পিউটার বিজ্ঞানের এই যুগটায় নয়মানের তুলনা বোধহয় শ্যানন আর ট্যুরিং ছাড়া আর কারো সঙ্গেই করা যায়না।

১.৩।। মশলে, একার্ট, এনিয়াক

এনিয়াক (ENIAC — **E**lectronic-**N**umerical-**I**ntegrator-**A**nd-**C**omputer) ছিল মার্ক ওয়ানেরও বাড়া, ওজনে তিরিশ টন। এতে ছিল ১৮০০০ ভালভ বা ভ্যাকুয়াম টিউব। চালানোর সময় বিদ্যুৎ লাগত ১৭৪ কিলোওয়াট এবং জায়গা লাগত পঞ্চাশ ফুট বাই তিরিশ ফুট। শ্যানন কাজ করেছিলেন যে ডিফারেনশিয়াল অ্যানালাইজারে, তিন নম্বর দিনে, লজিক সার্কিট উদ্ভাবনের সূত্রে আমরা যে মেশিনের কথা বলেছিলাম, তার থেকেই, তার উত্তরাধিকারী ছিল এই এনিয়াক। যুদ্ধের আগে, মূলত দূরপাল্লার গোলন্দাজির কাজে সহায়তার কারণে এনিয়াক বানানো হয়েছিল। এই সময়ের এই কম্পিউটার প্রকৌশলের এবং গোলন্দাজি বিজ্ঞানের গোড়ার দিককার সময় নিয়ে কিছু মজার গল্প আছে রিচার্ড ফেইনম্যানের আত্মজৈবনিক লেখা ‘শিওরলি ইউ আর জোকিং মিস্টার ফেইনম্যান’ বইটাতে, দু-একটা দেওয়াও যেত, কিন্তু জিএলটির কেয়া বইটা নিয়ে গেছে, কাছে নেই। ফেইনম্যান এই সময় কিছুদিন অর্ডন্যান্সে চাকরি করেন, সেই সময়টার নানা মজার অভিজ্ঞতা।



কামান থেকে গোলা ছোঁড়ার সময়, শেষ অব্দি কতটা দূর যাবে একটা গোলা, সেটা নির্ভর করে কতটা জোরে ছোঁড়া হচ্ছে, কত ডিগ্রিতে বেঁকানো আছে নলটা, গোলাটার আকার আকৃতি এগুলো ছাড়াও আবহাওয়ার অবস্থা তখন কী, বাতাসের গতি, বাতাসে আর্দ্রতা কতটা, সমুদ্রপৃষ্ঠ থেকে কতটা উঁচুতে আছে কামানটা, বায়ুমণ্ডলের তাপমাত্রা কী, এবং ভূমির উঁচুনিচু ইত্যাদি নানা কারণে সেখানে অভিকর্ষের অবস্থা — এইরকম অনেককিছুর উপর। গণিতের ভাষায় একে বলে প্যারামিটার বা নিয়ন্তা। এবার ভাবুন, একটা কামান মানেই কামানের সঙ্গে বিভিন্ন ধরনের মানের তালিকায় ভর্তি একটা বই। গোলা ছোঁড়ার সময় এই সমস্ত প্যারামিটার বা নিয়ন্তাগুলোর মানের তালিকা, তার সঙ্গে মিলিয়ে গোলা পৌঁছনোর সম্ভাব্য দূরত্ব। সভ্যতার হদ্দমুদ্দ একেবারে, সাক্ষর গাণিতিক সুচারু খুনোখুনি। বই পড়ো, গুলি মারো। কামান বদলানো গোলা বদলানো মানেই বই বদলানো।

আমেরিকায় এই গোলাবিজ্ঞানের কাজটা, প্যারামিটার সহ সম্ভাব্য দূরত্বের তালিকার বই লেখা যার একটা অংশ, প্যারামিটারের মানগুলো খুঁজে পাওয়া যার মূল গবেষণা — এর প্রায় গোটাটাই হত মেরিল্যান্ডের আবেরডিনে আর্মি অর্ডন্যান্স ব্যালিস্টিক রিসার্চ ল্যাবরেটরিতে। নতুন যুদ্ধের প্রস্তুতি মানেই রাশি রাশি নতুন হিশেব এবং নতুন বই। হিশেবের কাজ করার জন্যে ছিল ডিফারেনশিয়াল অ্যানালাইজার বা হাতে ডান্ডা ঘোরানো বিভিন্ন যান্ত্রিক ক্যালকুলেটর, ফ্যাসিট ইত্যাদি, যেগুলোর কথা আগেই বলেছি, বা একদম বাংলা রকমে হাতে করে। কিন্তু এইভাবে এসব করা মানেই খুব কম হলেও, একটা নতুন সেটের জন্যে বেশ কয়েক মাস লেগে যাওয়া। এই কাজের মূল দায়িত্বটা তাই দেওয়া হল পেনসিলভানিয়া বিশ্ববিদ্যালয়ে মুর স্কুল অফ ইলেকট্রিকাল ইঞ্জিনিয়ারিং-এর উপর। মুর স্কুলে এই সময় ছিলেন জন উইলিয়ম মশলে (১৯১৯-১৯৮০) এবং জন প্রস্পার একার্ট (১৯১৯-১৯৯৫)।

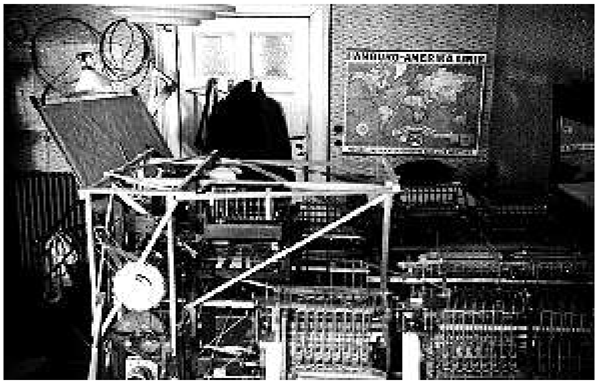
আবহাওয়া বিদ্যা এবং আবহাওয়ার পূর্বাভাষের ব্যাপারে এর আগে থেকেই আগ্রহী ছিলেন মশলে। সেই কাজে মূল প্রতিবন্ধকই ছিল বড় বড় একঘেয়ে শেষহীন জাম্বো হিশেব। তাদের সহজে চটকানোর স্বার্থে কম্পিউটারের বিষয়ে আগে থেকেই আগ্রহী ছিলেন মশলে। এই রকম সময়ে মশলের কানে আসে আইওয়া স্টেট কলেজে এবিসি বা

আটানাসফল্ট বেরি কম্পিউটারের কথা। '৪১-এ গিয়ে এই এবিসি দেখে এবং শিখে আসেন মশলে। এ বি সি শেখার পর এবার আর বাক্য লিখতে কতক্ষণ, বিয়াল্লিশের বসন্তে মশলে প্রস্তাব করলেন একটা বড় মাপের ইলেকট্রনিক ডিজিটাল কম্পিউটারের। কিন্তু বিশ্ববিদ্যালয়ের দাদারা, বিগ বসরা, এতে মুগ্ধ হলেন না, এবং ধামাচাপা পড়ে গেল এই প্রোজেক্ট। শেষ অব্দি, ফের প্রস্তাবের পর, '৪৩-এ এসে কাজ শুরু হল এনিয়াকের। মূল ইঞ্জিনিয়ার একার্ট এবং বিশেষজ্ঞ মশলে। পুরোদস্তুর কাজের মত মেশিন হয়ে এনিয়াক দাঁড়িয়ে উঠতে লেগে গেছিল '৪৬ অব্দি। মোগাম্বো খুশ হয়েছিলেন।

তার সময়ের অন্য মেশিনের তুলনায় এনিয়াকের গতি ছিল অবিশ্বাস্য রকমের বেশি। যোগ করতে লাগত ২০০ মাইক্রোসেকেন্ড। এক সেকেন্ডের দশলক্ষ ভাগের এক ভাগে হয় এক মাইক্রোসেকেন্ড। গুণ করতে লাগত ২.৬ মিলিসেকেন্ড, মিলিসেকেন্ড মানে এক সেকেন্ডের এক হাজার ভাগের এক ভাগ। বর্গমূল বার করতে লাগত ২৫ মিলিসেকেন্ড। বর্গমূল বার করায় এই বাড়তি গড়িমসির কারণ কোনো সংখ্যার বর্গ বার করার যে বিশেষ অ্যালগরিদমটা এনিয়াক ব্যবহার করত সেটার বিদঘুটেপনা। তাতে শুধু যোগ আর বিয়োগ করেই কাজ সারা যেত বটে, কিন্তু বড় বড় সংখ্যার বেলায় যে পরিমাণ যোগ আর বিয়োগ করতে হত সেটা রীতিমত অশ্লীল। অ্যালগরিদম বা ক্রিয়াপদ্ধতিটাও খুব মজার। কিন্তু নাম্বার সিস্টেমটা আগে একটু বলে না-নেওয়ায় সেটায় যেতে পারলাম না।

আর মূল যে কাজের এনি আঁক কষে দেওয়ার জন্যে এনিয়াককে এনে ফেলা হয়েছিল এই ধরাধামে, সেই যুদ্ধু ডিপার্টমেন্টের মহাপ্রভুরাও খুশি হয়েছিলেন তাদের এই ওজনদার দাসের কাজকর্মে। সবকটা নিয়ন্তা বা প্যারামিটার, ওই যে প্রভাবক বিষয়গুলো আমরা একটু আগে বললাম, তাদের মানগুলো দিয়ে দেওয়ার পর একটা গোলার গতিপথ বা ট্রাজেক্টরি কষতে এনিয়াকের লাগত তিরিশ সেকেন্ড, যে কাজে শ্যাননের ডিফারেনশিয়াল অ্যানালাইজারের লাগত পনেরো মিনিট, ক্যালকুলেটর হাতে একটা মানবকের যাতে লাগবে কমসেকম বিশ ঘন্টা। এদিকে গোলাটার গড়ে এক মিনিট লাগত ওই দূরত্বটা পেরিয়ে তার চাঁদমারিতে পৌঁছতে, তার মানে গোলার গতির চেয়ে গোলার গতির হিশেবের গতি ছিল এনিয়াকের। তাই এনিয়াককে ভালোবেসে বলাই যায়, গতিতে তুমি বজ্রতর, হে এনিয়াক। এনিয়াক কাজ করেছিল ১৯৫৫ অব্দি। আশি হাজার দুশো তেইশ ঘন্টা কাজ করার পর, হয়তো গান্ধীর অহিংসায় দীক্ষিত হওয়ার আশাতেই, ওয়ার ডিপার্টমেন্টের এই মেশিন শেষ বারের মত অফ করে দেওয়া হয়, ১৯৫৫-র ২ অক্টোবর, গান্ধীজয়ন্তীর দিন।

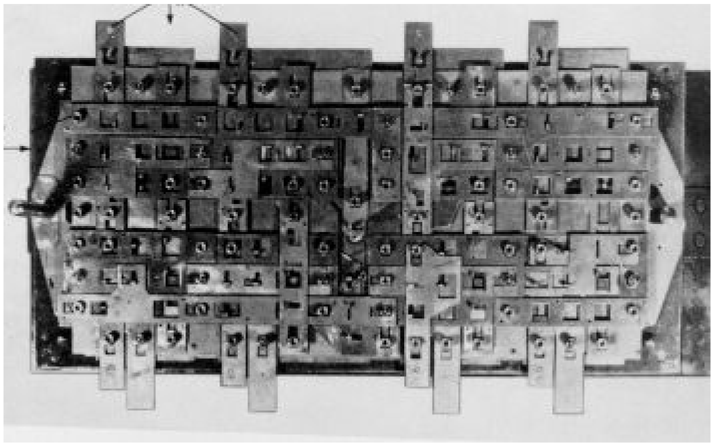
১.৪।। কনরাড জুসে, জেড সিরিজ, প্লানকালকুল



নিজের ঘরে বানিয়ে তোলা জেড ওয়ান

এর আগে যে তিনটে উদাহরণ দিলাম, হার্ভার্ডে আইকেন, প্রিন্সটনে নয়মান, পেনসিলভানিয়ায় মশলে — এই তিনটেই আমেরিকায়। এই প্রথম যুগের কম্পিউটারগুলোয় আমেরিকার বাইরে একটাই উদাহরণ, জার্মানিতে কনরাড জুসে (জন্ম বাইশে জুন, ১৯১০)। তাই, তুলনামূলক ভাবে অনেক কম উল্লেখ বা প্রচার হয় জেড-সিরিজের, পৃথিবীর মুখের ভাষা আর বুকের চিন্তাকে নির্ণয় করে দেয় আমেরিকা। অথচ, সেদিক দিয়ে দেখলে প্রথম কম্পিউটারের সম্মান পাওয়া উচিত জুসের, কারণ জুসের জেড সিরিজের প্রথম মেশিন জেড ওয়ান তৈরি হয়েছিল ১৯৩৬ থেকে ১৯৩৮-এর মধ্যে। আর আইকেনের মার্ক ওয়ান তৈরি হতে লেগেছিল ১৯৩৯ থেকে ৪৪।

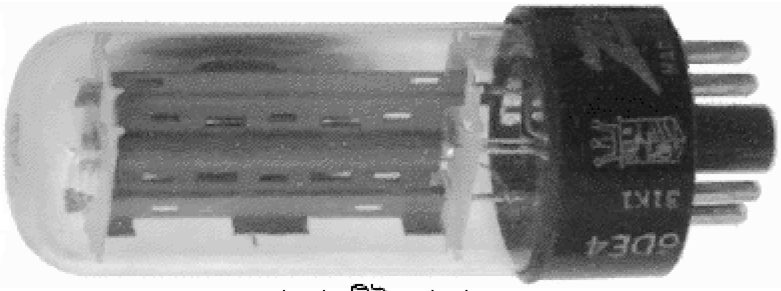
জুসে জেড ওয়ান বানান নিজেদের বসার ঘরে, প্রতিটি শব্দ ২২ বিটের এমন ৬৪-টি শব্দ নিজের স্মৃতিতে রাখতে পারত জেড ওয়ান। একে প্রোগ্রাম করা যেত এবং কাজ করত বাইনারিতে। জেড ওয়ান মেশিনে কোনো বৈদ্যুতিক রিলে ব্যবহৃত হয়নি, সেই কাজগুলো গোটাটাই করা হয়েছিল ধাতুর পাত জুড়ে জুড়ে, ছবিতে দেখুন। জিগ-স করাত দিয়ে কেটে কেটে জুসে আর তার বন্ধুরা নিজেদের হাতে বানিয়েছিল এই পাতগুলো। একমাত্র বৈদ্যুতিক যন্ত্রাংশ ছিল একটা মোটর যার দৌলতে একটা এক হার্জের ক্লক-ফ্রিকোয়েন্সি তৈরি হত মেশিনে। প্রোগ্রাম ঢোকানো হত একটা পাঞ্চ টেপ আর পাঞ্চ টেপ রিডার দিয়ে। শূন্য নম্বর দিনে আমাদের আনা সেই সরলতম আর্কিটেকচারের নিরিখে ভাবতে গেলে, পরিস্কার আলাদা করা যেত নিয়ন্ত্রণ উপাদান বা কন্ট্রোল ইউনিটকে এই পাঞ্চ টেপ রিডার থেকে। আলাদা করা যেত এর অ্যারিথমেটিক ইউনিট বা প্রসেসিং অংশ এবং স্মৃতি বা মেমরিকেও এবং ইনপুট-আউটপুট ডিভাইসকেও। চমৎকার করে এই গোটাটা বোঝানো আছে কনরাড জুসের ছেলে হোর্স্ট জুসের লেখা বই ‘দি লাইফ অ্যান্ড ওয়ার্ক অফ কনরাড জুসে’ নামে একটা বইয়ে, গোটা বইটাই নেটে পাওয়া যায়। www.epemag.com সাইট থেকে। কনরাড জুসের এই জেড সিরিজে জেড টু জেড থ্রি ইত্যাদি আরো সংযোজন এসেছিল। গোটা গল্পটা বেশ ইন্টারেস্টিং।



পাতলা ধাতুর পাত জুড়ে জুড়ে তৈরি হয়েছিল হিশেব করার যন্ত্রপাতি

যুদ্ধের সময় বোম পড়ে সম্পূর্ণ নিশ্চিহ্ন হয়ে যায় জেড ওয়ান। কিন্তু নাজি সরকার তার কাজের কোনো পৃষ্ঠপোষকতা করেনি, জুসে পালিয়ে আসেন জুরিখে, জেড-থ্রির কিছু খণ্ডাংশ নিয়ে। শুধু কম্পিউটারই নয়, একটা ভীষণ প্রাথমিক স্তরের অপারেটিং সিস্টেম কাম হাইলেভেল ল্যাংগুয়েজও বানান জুসে, তার নাম ‘প্লানকালকুল’। ডেসিমাল বা দশমিক পদ্ধতিতে সাতাশটা অব্দি সংখ্যা নিয়ে কাজ করতে পারা আডা-ব্যাবেজের অসমাপ্ত অ্যানালিটিকাল ইঞ্জিনের কথা জুসে জানতেন না। স্বতন্ত্র ভাবে জুসে নিজেই একটা প্রোগ্রামিং কাঠামো ভেবে বার করেন, যার ফলাফল প্লানকালকুল।

ভ্যাকুয়াম টিউব বা ভালভের ভিতর থাকে এক সেট ধাতব ইলেকট্রোড বা তড়িৎদ্বার, আর এদের সবাইকে ধরে রাখার আর বিদ্যুৎ যোগাযোগের জন্যে একটা তারের কাঠামো বা গ্রিড। এই গোটাটা ভরা থাকে একটা বায়ুশূন্য কাঁচের বা ধাতুর টিউবে।



ভ্যাকুয়াম টিউব বা ভালভ

গ্রিডের শরীরে ভোল্টেজ বা তড়িৎ-বিভবের মানই ইলেকট্রোডগুলোর ভিতর তড়িৎপ্রবাহকে নিয়ন্ত্রণ করে। ভালভ ব্যবহার করা হত অ্যাম্পলিফিকেশন বা পরিবর্ধনের কাজে, বা সার্কিটে তড়িৎপ্রবাহ নিয়ন্ত্রণের কাজে, এখনও ব্যবহার হয়, ক্যাথোড রে

টিউবে, বা ভয়ানক উঁচু মাত্রার বিদ্যুৎ যেখানে ব্যবহার হয়। এর মার্কিন নাম ভ্যাকুয়াম টিউব, ব্রিটিশ নাম ভালভ। আমাদের, মেকলের অবৈধ বাচ্চাদের, দুটোই জানার কথা।

২।। প্রথম জেনারেশন কম্পিউটার — ১৯৪৫-৫৫

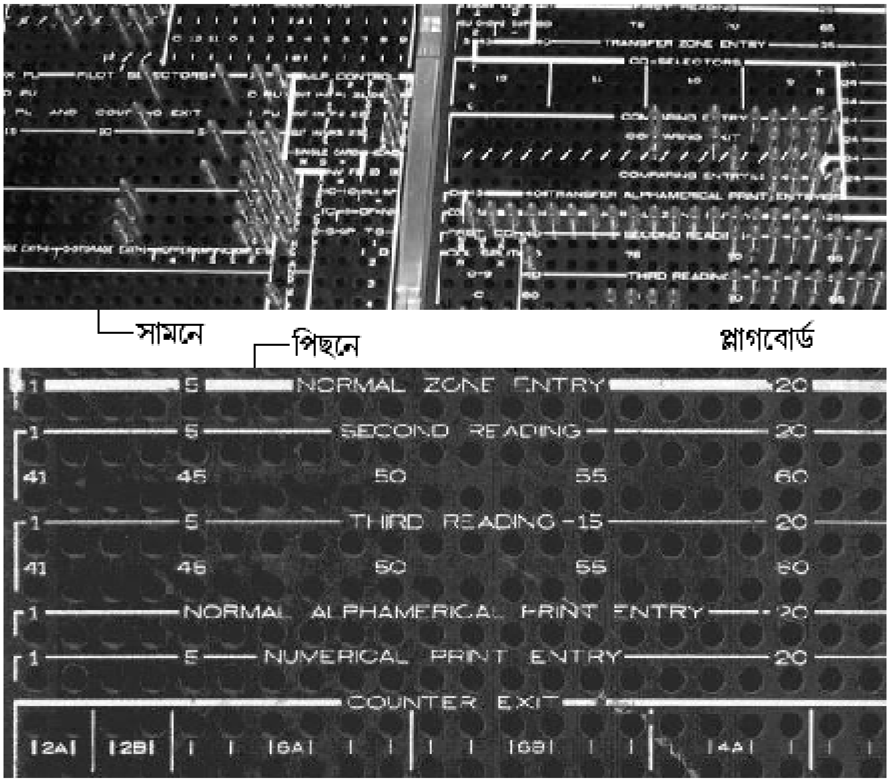
এই আইকেন জুসে নয়মান মশলেদের কাঁধে ভর করে পৃথিবীতে দেখা দিলেন আধুনিক ডিজিটাল কম্পিউটারের প্রথম প্রজন্ম। বা ফার্স্ট জেনারেশন। এই স্তরের কম্পিউটারগুলোকে এক কথায় ডাকা যায় ভ্যাকুয়াম টিউব আর প্লাগবোর্ডের মেশিন বলে। তখনো ইলেকট্রনিক রিলে ব্যবস্থা আসেনি — শ্লথগতি যান্ত্রিক রিলেই ব্যবহৃত হয়েছিল। পরে এই রিলেগুলোর জায়গায় আনা হয় ভ্যাকুয়াম টিউব। এই যুগের কম্পিউটারগুলোর আকার ছিল বিপুল। মার্ক ওয়ান বা এনিয়াক গোছের এদের পূর্বপুরুষদের সাইজ তো এইমাত্রই দেখলেন। আস্ত আস্ত ঘর হলঘর ভরে যেত এক একটা কম্পিউটারের ভিতরের হাজার হাজার ভ্যাকুয়াম টিউবে। ভ্যাকুয়াম টিউব থেকে পরবর্তী যুগের ট্রানজিস্টরে রূপান্তর যেমন একটা নাটকীয় বদল, এর আগের অবস্থান থেকে ভ্যাকুয়াম টিউবও একটা নাটকীয় বদল ছিল।

অথচ এই সময়কার মেশিনগুলোর এই জগদ্দল আয়তনের পরেও, আজকের পৃথিবীর সবচেয়ে সস্তা কম্পিউটারও এদের চেয়ে অযুত গুণ দ্রুত চলে। এই সময়ের কম্পিউটারগুলো চলত আজকের আমাদের কাছে সম্পূর্ণ অপরিচিত এক কায়দায়। এক একটা কম্পিউটার মানেই তখন এক এক সেট লোক — যারা মেশিনটা বানিয়েছে, যারা মেশিনটা চালায়, যারা প্রোগ্রাম করে এবং যারা গোটা ব্যাপারটার রক্ষণাবেক্ষণ করে। এদের বাইরে অন্য সবার কাছে কম্পিউটার তখন একটা অবোধ্য জগত। এই মেশিনের প্রোগ্রামগুলোও লেখা হত মেশিন ল্যাংগুয়েজে। এমনকি অ্যাসেম্বলি ল্যাংগুয়েজও তখন আসেনি।

অ্যাসেম্বলি ভাষা মানে একদম নিম্নতম স্তরে কম্পিউটার আর্কিটেকচারকে নির্দেশ দেওয়ার জন্যে তৈরি এক ধরনের একটা ভাষা, যেমন অনেকটা উপরের স্তরের একটা ভাষা হল সি। একটা ভাষা মানে কিছু বিশেষ অর্থবাচক শব্দ এবং কিছু নিয়মের সমষ্টি। বাংলা ইংরিজি যে কোনো ভাষা মানেই তাই। এই বললে আমি এই বুঝব, ওই বললে ওই, এবং এই এই জিনিষকে এই এই ভাবে বলা যাবে। কিন্তু বাস্তব জীবনের যে কোনো ভাষার একটা ইতিহাস থাকে, কী ভাবে এই বিশেষ অর্থবাচক শব্দ আর বিশেষ নিয়মগুলো গজিয়ে উঠল। ব্যক্তি মানুষের কোনো একক বা সমষ্টি সেখানে সিদ্ধান্ত নেয়না, সিদ্ধান্ত নেয় জ্যান্ত ইতিহাস। কিন্তু কম্পিউটার ভাষারা সেই অর্থে নির্মিত। বিশেষ কিছু কাজের উদ্দেশ্যে বানিয়ে তোলা হয়। সেখানে উদ্দেশ্যটা থাকে এমন ভাবে অর্থপূর্ণ শব্দের তালিকা আর নিয়মগুলো বানিয়ে তোলা, যাতে কাজগুলো করা সহজ হয়। যাতে, এই নিয়মগুলোকে ব্যবহার করে ক্রমে জটিল জটিলতর নির্দেশও সহজে দেওয়া যায়। আমাদের রোজকার ব্যবহারের ভাষার সঙ্গেও এই অর্থে একটা কাঠামোগত মিল আছে এর। কম্পিউটার ভাষার এক একটা কি-ওয়ার্ড হল আমাদের ভাষার এক একটা জার্গনের মত। ধরুন, এই যে আমরা এখন মুর্হূমুহু 'কিবোর্ড' শব্দটা ব্যবহার করতে পারছি কারণ ওই শব্দের পিছনে গোটা অর্থ বা ধারণা বা ছবিটা আমাদের কাছে প্রদত্ত। এবার এদের ব্যবহার করে জটিলতর ধারণাগুলোকেও আমরা পৌঁছে দিতে পারছি। এর আবার কিছু নিয়ম আছে, যেমন, আমরা কিবোর্ড সরিয়ে রাখি বা চাবি-টিপি কিন্তু কাপে ঢেলে রাখিনা বা উড়িয়ে দিই-না। কিন্তু খেয়াল করুন, এখানে পারা বা না-পারাটা বাস্তবতা থেকে এসেছে, আমাদের কিছু করার নেই, একটা চৌকো জড় পদার্থকে ঢালা বা ওড়ানো যায়না, কিন্তু কম্পিউটার ভাষার বেলায় এটাও বানিয়ে তোলা, ভাষা উদ্ভাবক বা প্রোগ্রামচিরা মিলে ঠিক করে নেওয়া। ঠিক একই ভাবে, সি ভাষার কিছু কি-ওয়ার্ড আছে, যেমন, শূন্য নম্বর দিনে আমাদের সেই গাধা-বাচক প্রোগ্রামে 'int', 'return' ইত্যাদি। কিছু নির্দিষ্ট নিয়ম আছে, যা মেনে নির্দেশগুলোকে লিখতে হয়। নিয়মও আছে, ওই কোডটার একটা ';' তুলে নিলেও কোডটা আর ব্যবহার করা যেতনা।

একটা কম্পিউটারে একটা ভাষা ব্যবহার করা যায়, মানে, ওই ভাষার কি-ওয়ার্ডগুলো, নিয়ম-কানুন, ব্যাকরণ মেশিনটার মধ্যে ভরা আছে। সেই নিয়মকানুন মেনে কথা বললে, কী বললে কী করতে হবে এই সরলতর নির্দেশগুলো তার মধ্যে আগে থেকেই ভরা আছে। এই অ্যাসেম্বলি ভাষা আবার এতটাই সরল সাদামাঠা এবং আপাতদৃষ্টিতে অঙ্কের লাইনের মত দেখতে যে তাকে ভাষার চেয়ে বাচন বলাই শ্রেয়। সেই অ্যাসেম্বলি ভাষাও ভরা

ছিলনা এই যুগের এই মেশিনগুলোয়। প্রোগ্রামিং ল্যাংগুয়েজ বা অপারেটিং সিস্টেম — ওসবের কেউ নামই শোনেনি। কম্পিউটারকে ন্যূনতম আদেশটাও দিতে হত একদম সরাসরি নিম্নতম আর্কিটেকচার যে আদেশ বুঝতে এবং পালন করতে পারে সেই আদেশের এককে।



একটা প্রোগ্রাম চালানো মানে প্রোগ্রামচি এসে একটা নির্দিষ্ট সময়ের জন্যে বুক করে নিত মেশিনটাকে, তারপর চলে যেত সেই রুদ্ধ কক্ষে — বিনা কারণে, বিনা অনুমতিতে যেখানে প্রবেশ নিষেধ, যেখানে তিনি বিরাজ করেন — তেনার নাম কম্পিউটার। এরপর মেশিনের জটিল দেহে ঢুকিয়ে দিতে হবে সেই প্লাগবোর্ড। চল্লিশ এবং পঞ্চাশের দশকে, বিশেষ করে নানা ধরনের হিশেবের কাজ, সে ব্যবসার হিশেবই হোক বা বিশুদ্ধবিজ্ঞানের গণিতের, একধারসে হয়েছে এই প্লাগবোর্ড দিয়ে। এই ব্যবস্থাটা কিন্তু, ইলেকট্রনিক নয়, ইলেকট্রিকাল। কার্ডের গায়ে নানা তারের কানেকশন জুড়ে জুড়ে তৈরি হত প্লাগবোর্ড। বোর্ডের গায়ে থাকত সকেট, তার মধ্যে কেবলগুলো গুঁজে দেওয়া হত। এই প্লাগবোর্ডই ধারণ করত এখুনি চালানোর প্রোগ্রামটাকে। প্লাগবোর্ড গুঁজে দিত এবং ইষ্টনাম জপ করত প্রোগ্রামার। কম্পিউটারের মধ্যে ভরা হাজার বিশেক ভ্যাকুয়াম টিউব বা ভালভের একটাও যদি বিগড়োয় তাহলেই গেল। গোটা কাজটাই হত সরাসরি হিশেবের — রাশি রাশি সাইন, কস, ট্যান, বা লগারিদম, বা কোটি কোটি যোগ বিয়োগ। তার মানে, আমাদের তিন নম্বর দিনের আলোচনার সঙ্গে মিলিয়ে বলা যায়, শুধু ব্যাবেজের অ্যানালিটিকাল ইঞ্জিনে যা হত গিয়ার গুলি ডান্ডা দিয়ে, মানে মেকানিকাল উপায়ে, এখানে সেই কাজটা হচ্ছে লজিক রিলে আর ভালভ দিয়ে, মানে ইলেকট্রনিকালি। কিন্তু কম্পিউটার মননের স্তরে তেমন কোনো বদল তখনো আসেনি।

১৯৫০-এর গোড়ার দিকে এল ওই পাঞ্চড কার্ড, যার কথা আগেই বলেছি আমরা। কার্ডের গায়ে ফুটো করে করে কম্পিউটারকে কাজের নির্দেশ এবং সেই কাজের জন্যে প্রয়োজনীয় তথ্য দেওয়া। তিন নম্বর দিনে, যে কার্ড নিয়ে বেশ কিছু কথা বলেছি আমরা। পাঞ্চড কার্ড আসায় প্লাগবোর্ড অচল হয়ে গেল। কিন্তু এইটুকু বদলকে ছেড়ে দিলে মেশিনগুলো বা তাদের কাজের পদ্ধতিটা রয়ে গেল প্রায় একই।

৩।। দ্বিতীয় জেনারেশন কম্পিউটার — ১৯৫৫-৬৫

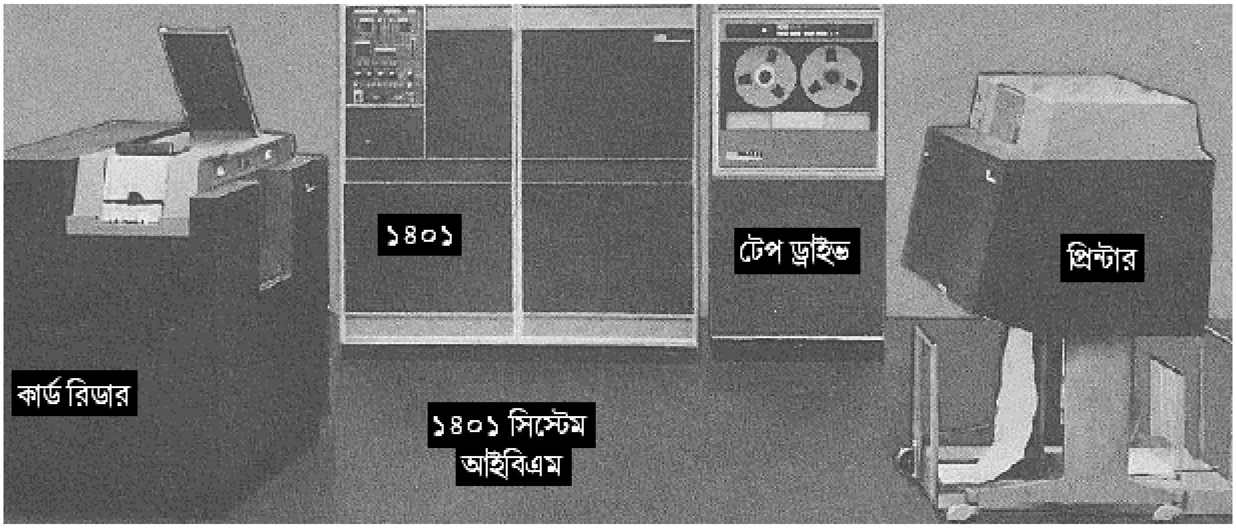
এবার ভ্যাকুয়াম টিউবের দিন শেষ। এল ট্রানজিস্টর। এবং তার সঙ্গে এল ব্যাচ সিস্টেম। মানে একসাথে তালিকাবদ্ধ অনেকগুলো কাজ পর পর করার ব্যবস্থা। পরে, আমাদের এই পাঠমালায় আবার আমরা দেখব, এবং আগেও উল্লেখ

করেছি, এই ব্যাচ সিস্টেম ব্যাপারটার মধ্যে কিন্তু প্রোগ্রামিং ক্রিয়ার বেশ কিছু চিন্তন উপাদান উপ্ত হয়ে আছে। পরে, শেল স্ক্রিপ্টিং-এর প্রসঙ্গে আমরা ফেরত আসব এই কথায়।

বাজারে ট্রানজিস্টর এসে গিয়ে পুরোটাকে একসঙ্গে অনেকটা বদলে দিল। কম্পিউটার ব্যাপারটাই আগের চেয়ে অনেক নির্ভরযোগ্য হয়ে উঠল — যাতে বানিয়ে লোককে বেচা যায়, এতদিন কম্পিউটারগুলো চলত অহর্নিশি নির্মাতার তত্ত্বাবধানে, এখন সেটা ক্রমে কমতে শুরু করল। এতদিনকার ওই কম্পিউটার পিছু এক সেট মানুষের মধ্যে এবার ভাগ তৈরি হল। কেউ কম্পিউটারের ডিজাইন তৈরি করে — ডিজাইনার, কেউ বানায় — বিল্ডার, কেউ চালায় — অপারেটর, কেউ প্রোগ্রাম লেখে — প্রোগ্রামার, কেউ দেখভাল করে — মেইনটেইনার। এতদিন অব্দি এদের প্রত্যেককেই প্রায় প্রত্যেকটা কাজই করতে হত।

এই মেশিনগুলোর চালু নাম এল — মেইনফ্রেম। চাবি-লাগানো দরজার ওপারে বাতানুকূল কামরায় নির্দিষ্ট কিছু অপারেটর তাকে চালায়। আর সেই জগদ্দলের দাম যা তাতে সরকার বা বিশ্ববিদ্যালয় ছাড়া কারুর পক্ষে কেনা অসম্ভব। যে কোনো কাজ, সে একটা প্রোগ্রামের হোক, বা অনেকগুলো, প্রোগ্রামারকে প্রথমে সেটা কাগজে কলমে লিখে ফেলতে হবে, হয় অ্যাসেম্বলি কোডে, অথবা ফোর্ট্রান নামের কম্পিউটার ভাষায় (FORTRAN — **FOR**mula-**TRAN**slation)। বিজ্ঞানের কাজের জন্যে তৈরি কম্পিউটার ভাষা ফোর্ট্রান তদ্দিনে এসে গেছে। বানিয়েছেন জন ব্যাকাস, ১৯৫৪ থেকে ৫৮-য়। পরবর্তীকালের হাইলেভেল কম্পিউটার ভাষাগুলোর প্রচুর ধারণা এসেছে এই ফোর্ট্রান থেকে। বিজ্ঞান এবং ইঞ্জিনিয়ারিং-এর কাজে এখনো ব্যাপকভাবে ব্যবহার হয়।

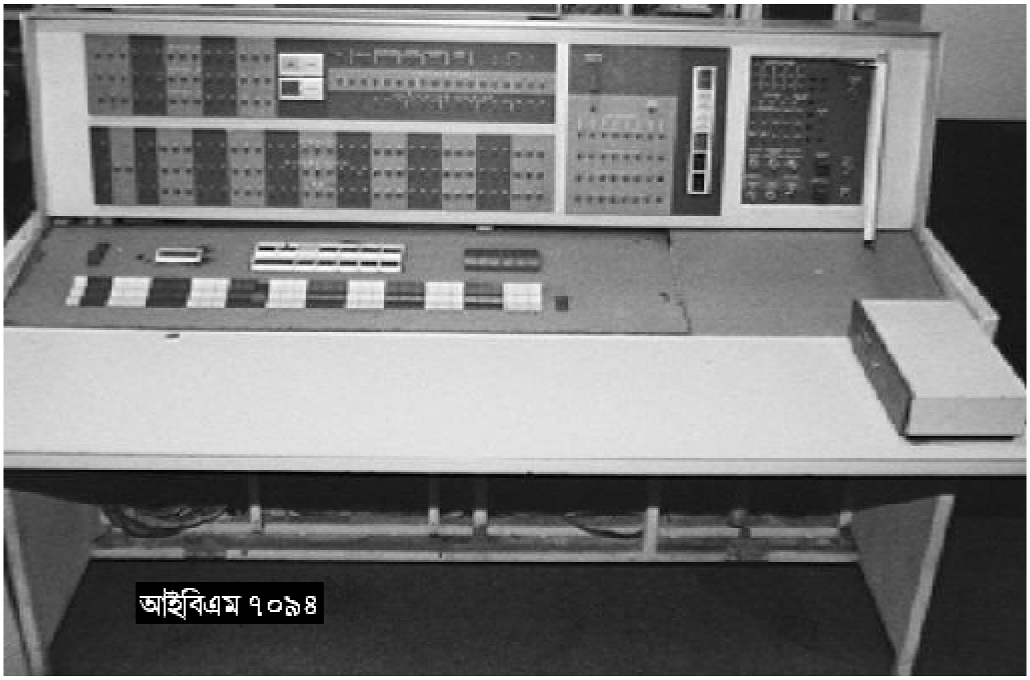
এর পর, কাগজে-কলমে লেখা প্রোগ্রামটাকে নিয়ে যেতে হত কার্ড লেখার মেশিনে। কার্ড-লেখার মেশিন দিয়ে এই এই প্রোগ্রামটাকে পাঞ্চড কার্ডের গায়ে ফুটোর অক্ষরে লিখে দেওয়া হত। ফুটো-বিভূষিত সেই কার্ডমালা তারপর নিয়ে আসা হত সর্বসাধারণের প্রবেশ-অযোগ্য সেই কম্পিউটার কক্ষে, অপারেটরের দায়িত্বে। প্রোগ্রামারের তখন ছুটি। এখন থেকে দায়িত্ব অপারেটরের, আগের প্রোগ্রামের চলতে থাকা কাজ শেষ হলে পুরোনো কার্ডমালা বার করে অপারেটর নতুন প্রোগ্রামের নতুন সেট গুঁজে দেবে কম্পিউটারের পেটে। যদি প্রোগ্রামটা লেখা হয় ফোর্ট্রানে, তাহলে ফোর্ট্রান কম্পাইলারের নিজস্ব কার্ডমালাও যোগ করে দিতে হবে সঙ্গে, আগেই তো বললাম, আদেশ বোঝার ভাষা কম্পিউটারকে বোঝাতে গেলে তার জন্যে দিতে হয় সরলতর আদেশ, সেটাই কম্পাইলারের কাজ।



কম্পিউটারের বেশির ভাগ সময়টাই নষ্ট হত অপারেটারদের কার্ড ঢোকানো বার-করা, আর একটা প্রোগ্রামের পর আর একটা প্রোগ্রামের ফাইল নিয়ে আসার কাজে। আর নষ্ট করার পক্ষে কম্পিউটার-সময়ের দামটা ছিল বড্ড বেশি। তাই এল ব্যাচ-সিস্টেম। যাতে পর পর অনেকগুলো কাজ একসঙ্গে তুলে দেওয়া হত ম্যাগনেটিক টেপে। এই তুলে দেওয়ার কাজটা হত তুলনামূলক ভাবে অনেক সস্তা একটা কম্পিউটারে। এই ধরনের কম্পিউটারের একটা ভালো উদাহরণ আইবিএম ১৪০১। ১৪০১ মেশিনটা এই কাজের জন্যেই তৈরি। যাতে অজস্র কার্ড পরপর খুব তাড়াতাড়ি পড়ে ফেলতে পারে, এবং সেই সমস্ত কার্ডের যাবতীয় তথ্য এবং আদেশ এবং সেই আদেশ পালনের

জন্যে প্রয়োজনীয় সরলতর আদেশ — এই যাবতীয় তথ্য — মানে কমান্ড আর ডেটা এই দুই রকম ডেটা — ঝটাঝট লিখে ফেলতে পারে ম্যাগনেটিক টেপে। কিন্তু অন্য ধরনের কাজে, গাণিতিক হিশেবের কাজে অন্য উন্নততর কম্পিউটারের মত দক্ষ নয় ১৪০১।

তারপর, ঠিক যেমন একটা ক্যাসেট গানে ভরে গেলে রেকর্ড হওয়া ক্যাসেটটা আমরা ঘুরিয়ে শুরুতে নিয়ে যাই বাজানোর জন্যে, একটা নির্দিষ্ট সময় ধরে ১৪০১-গোছের কম্পিউটার দিয়ে যথেষ্ট পরিমাণ কমান্ড আর ডেটা টেপে জমা হয়ে গেলে, টেপটা রিওয়াইন্ড করা হত, মানে ঘুরিয়ে গোড়ায় নিয়ে যাওয়া হত, যাতে সেটা থেকে তথ্য এবং আদেশ পড়ার সময় শুরু থেকে শুরু করা যায়। আর ভর্তি হয়ে যাওয়া ইনপুট টেপটা নিয়ে যাওয়া হত আসল কম্পিউটারে, যে কম্পিউটার অঙ্ক কষার কাজটা, মানে আসল কাজটা অনেক বেশি গতিতে করতে পারে। এই পরের কম্পিউটারটা অনেক বেশি দামী, যেমন আইবিএম ৭০৯৪। ওই তথ্য আর আদেশ ভরা টেপটা এই ৭০৯৪ কম্পিউটারে ঢোকানোর আগে অপারেটর এবার ঢোকাবে একটা বিশেষ প্রোগ্রাম, আজকের অপারেটিং সিস্টেমের কোনো আদি আদি পূর্বপুরুষ, যিনি কম্পিউটারকে বলে দেবেন ওই টেপে লেখা পরপর কাজগুলো কম্পিউটার কী ভাবে করে যাবে। এই আদি অপারেটিং সিস্টেমই এবার পড়তে শুরু করবে টেপটা এবং টেপে লেখা আদেশ মোতাবেক টেপে লেখা তথ্য চটকানোর কাজটা শুরু করে দেবে।



৭০৯৪ আবার যে কাজগুলো করবে তার ফলাফলগুলো লিখে রাখবে অন্য একটা টেপে, তার নাম আউটপুট টেপ। যেই একটা করে কাজ শেষ হবে, অপারেটিং সিস্টেম কাজের লাইনে বসে থাকা পরবর্তী কাজটা ঢুকিয়ে দেবে মেশিনে। কাজের পুরো ব্যাচটা হয়ে গেলে অপারেটর এবার সেই প্রাথমিক ইনপুট টেপ আর ৭০৯৪-এর তৈরি এই আউটপুট টেপটা দিয়ে দেবে প্রোগ্রামটা যে দিয়েছিল তার কাছে। প্রোগ্রামার আবার সেই টেপ নিয়ে ফেরত যাবে ১৪০১-এর কাছে। সে ওই টেপ থেকে ফলাফলটার প্রিন্টআউট বার করে দেবে তার সিস্টেমের প্রিন্টার দিয়ে।

আজ আমরা লিখছি, বা প্রোগ্রাম করছি, চালাচ্ছি, কমান্ড দিতে না দিতেই কাজ হয়ে যাচ্ছে, কাজ শেষ করা মাত্রই ফাইলটা প্রিন্টার পোর্ট এলপিটি-ওয়ানে (এই এলপিটি ওয়ান বা LPT1 — **L**ine-**P**rint-**T**erminal-numbered-1, লাইন- প্রিন্ট-টার্মিনালও সেই অর্থে একটা লুপ্ত মেটাফর, এখন আর এই ভাবে প্রিন্টই হয়না, নামটা চলে আসছে সেই তখন থেকে যখন এভাবে প্রিন্ট হত) কপি করার মানে প্রিন্ট করার আদেশ দিচ্ছি, বা গুইতে মানে ছবি আর মাউস দিয়ে কাজ করলে কেডিট বা ওওও-র (OOo — **O**pen**O**ffice.**o**rg) রাইটারের মেনুবারে প্রিন্টারের ছবি দেওয়া আইকনে ক্লিক করছি, খ্যাচখ্যাচ করে প্রিন্ট বেরিয়ে আসছে, আমাদের পক্ষে আজ চেষ্টা করলেও আন্দাজ করা শক্ত ওই সময়ের কম্পিউটার চালানোর সামগ্রিক ঝঞ্ঝাটটা। একটা জিনিষ খেয়াল করুন, শূন্য নম্বর দিনে আমরা

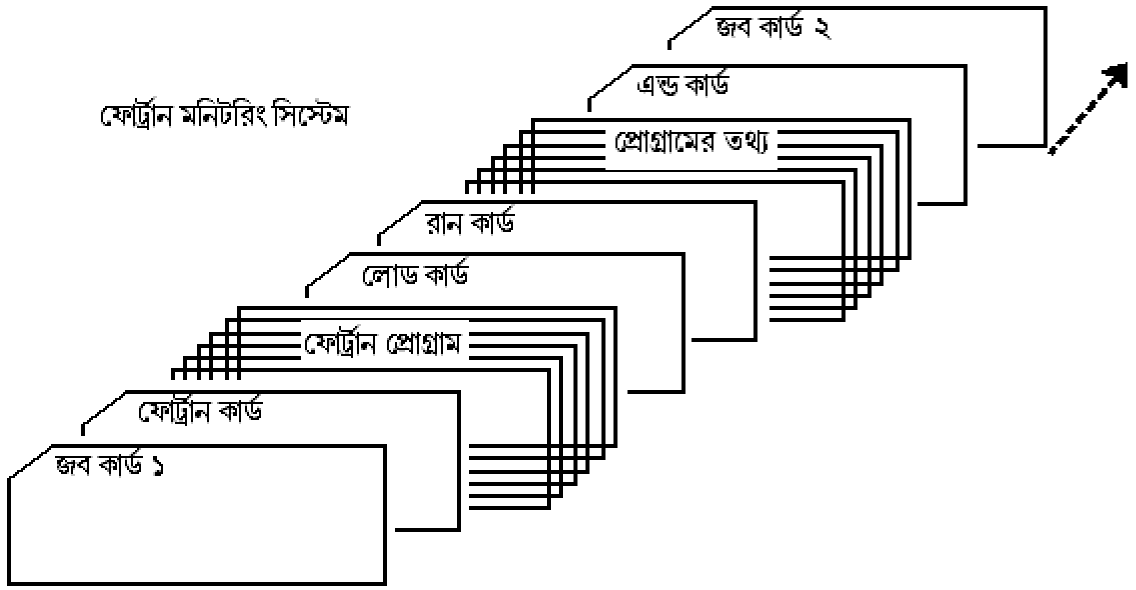
কম্পিউটারের যে কাজভিত্তিক ছকটা দিয়েছিলাম, সেটা কিন্তু এখানে সত্যি হচ্ছে একটা কম্পিউটারে নয়, আইবিএম ১৪০১ আর আইবিএম ৭০৯৪ এই দুটো মেশিনকে মিলিয়ে।

৪।। ফোর্ট্রান মনিটরিং সিস্টেম — অপারেটিং সিস্টেমের আদিপুরুষ

এই আদিতম অপারেটিং সিস্টেম তার বশংবদ ১৪০১ আর ৭০৯৪-কে দিয়ে কাজটা যে করাল, সেই কাজটার মধ্যে কিন্তু একটা খুব সুস্পষ্ট কাঠামো আছে। ১৪০১-এ সরাসরি হয়ত অপারেটিং সিস্টেম-এর কোনো অস্তিত্ব নেই, কিন্তু ৭০৯৪-এ যে কাজটা শেষ হচ্ছে, অপারেটিং সিস্টেম-এর তত্বাবধানে, তার শুরুটা হয়েছিল তো ১৪০১-এ তথ্য তোলা থেকে। সেই অর্থে তাই গোটা কাজটা ঘটছে দুটো মেশিনকে মিলিয়েই। আমাদের মধ্যমগ্রাম জিএলটির অন্তত দুজনকে দেখেছি এটা নিয়ে জিগেশ করতে — আদেশ আর তথ্য — দুটোই তো তথ্য। সেদুটোকে আলাদা করা হচ্ছে কী করে? আমরা এখানে গন্ডগোল পাকাচ্ছি কাজ শব্দটা দিয়ে। কাজটা একটা কাজ, আবার কাজটা পালন করাটাও কাজ। আবার, ইয়ে, অকাজও তো কাজ, আমাকে এত বছর ধরে দেখার পর মা এই সুস্পষ্ট সিদ্ধান্তে পৌঁছেছে — আমি 'কাজ' শব্দটা উচ্চারণ করলেই মুখটা কিরকম বেগুনভাজার মত (কার্টসি প্যালা) করে ফেলে।

তার চেয়ে ১৪০১ টেপে করে ৭০৯৪-এর কাছে যা পাঠাল তাকে ডাকা যাক 'জব' বলে। এবার ৭০৯৪-এর মোট কাজটাকে যদি কাজ বলে ডাকি, যা চলছে ওই আদি অপারেটিং সিস্টেমের অতন্দ্র তত্ত্বাবধানে, তাহলে সেই 'কাজ' কিন্তু তৈরি 'জব ১', 'জব ২' . . . ইত্যাদি পরপর টুকরোগুলোকে জুড়ে, কিন্তু শুধু ওই টুকরোগুলোই নেই, আরো কিছু আছে সেখানে। এখানে এই আলোচনাটা পরে অপারেটিং সিস্টেমের কাঠামো বুঝতে আমাদের খুবই কাজে লাগবে, বারবার, খেয়াল করে পড়ুন।

ধরুন ওই আদি অপারেটিং সিস্টেমের পাহারায় পরপর তার জবগুলোকে করে যাচ্ছে ৭০৯৪। এবার সে একটা জবকে জব বলে চিনছে কী করে? প্রথমে নিশ্চয়ই জব বলে চেনানোর একটা রকম থাকবে। রকম আর কী, সবই তো কার্ড দিয়ে হচ্ছে। তার মানে প্রথমে একটা কার্ড থাকবে যা বলে দেবে এবার আসছে এই কাজ। যে কার্ডে কী কাজ, কত বড়, কত তারিখ কী সময়ে কে দিয়েছে। এগুলো থাকত কম্পিউটার মালিক যাতে তার নিজের হিশেব রাখতে পারে — কে কতটা কাজ করাচ্ছে তার মেশিনকে দিয়ে। এবং, আগেই বলেছি, সেইসময় একটা মেশিনের যা দাম, একটা সংস্থার নিজের একটা কম্পিউটার খুব কমই থাকত। বেশিরভাগ ছোট বা মাঝারি মাপের কোম্পানি কোনো একটা বড় সংস্থার কম্পিউটার থেকে এই কাজগুলো করিয়ে নিত।



যাই হোক, কাজের খুঁটিনাটি লেখা এই কার্ডটার আমরা নাম দিলাম জব-কার্ড। এবার এই জব-কার্ডে দেওয়া কাজটা চালানোর জন্যে ফোর্ট্রান প্রোগ্রামটা লাগবে। তার মানে আবার একটা কার্ড, ফোর্ট্রান প্রোগ্রামটার নামপত্র, ধরুন এর নাম দিলাম ফোর্ট্রান-কার্ড। এইটা দেখে অপারেটিং সিস্টেম বুঝতে পারবে যে এই প্রোগ্রামটা চালাতে হবে ফোর্ট্রান

নামক কম্পাইলার দিয়ে। ফোর্ট্রান কম্পাইলার মানে ফোর্ট্রান ভাষায় লেখা প্রোগ্রাম চালাতে গেলে কম্পিউটারকে যে আদেশগুলো দিয়ে রাখতে হয়, আমরা আগেই বলেছি। অপারেটিং সিস্টেম এবার টেপ থেকে ফোর্ট্রান কম্পাইলারের আদেশগুলো পড়ে নেবে। টেপ থেকে এই পড়াটা হয়ে গেলে একটা আদেশ দিতে হবে এবার প্রোগ্রামটা চালানোর। কিন্তু, চালানোর আগে দুটো স্তরে কাজ সারতে হবে। প্রথম স্তরে মানুষবোধ্য ফোর্ট্রান ভাষায় লেখা কোডটাকে তুলে নিতে হবে কম্পিউটারে। তারপরে, দ্বিতীয় স্তরে সেটাকে কম্পাইল করতে হবে। মানে প্রোগ্রামারের হাইলেভেল ভাষায় লেখা বিশেষ আদেশমালাকে কম্পিউটারের কাছে যন্ত্রের ভাষায় বোধ্য করে তুলতে হবে। যেই কম্পিউটারবোধ্য করা হল, কম্পাইল করার কাজ শেষ হল। যন্ত্রে চালানোর মত প্রোগ্রাম প্রস্তুত হয়েছে। এবার চালানো যায় মেশিন।

কম্পাইল করা যন্ত্রবোধ্য প্রোগ্রামটাকে এবার মেশিনের চালু মেমরিতে তুলে নিতে হবে। মানে, আর একটা কার্ড। ধরুন সেটার নাম লোড-কার্ড। এবার আসবে সরাসরি কম্পাইলড প্রোগ্রামটা চালানোর আদেশ। ধরুন এই কার্ডটার নাম রান-কার্ড। এবার রান তো করবে মেশিন, কিন্তু কার উপর রান করবে? তার মানে, এই প্রোগ্রাম দিয়ে চটকানোর জন্যে প্রয়োজনীয় তথ্য। ধরুন একটা বড় অঙ্ক কষা হচ্ছে, কষতে হবে যে ফরমুলা দিয়ে সেটা তো প্রোগ্রাম করে আপনি ঢুকিয়ে দিয়েছেন, এবার কষতে হবে যে সংখ্যাগুলো দিয়ে সেগুলো ঢোকাতে হবে। শুধু অঙ্ক কষা হলে এই তথ্য মানে শুধু সংখ্যা বা নিউমেরিক তথ্য, অন্য কাজ হলে টেক্সট এবং অঙ্ক এই দুরকম মানে আলফানিউমেরিক তথ্য। তথ্যের এই রকমগুলো শূন্য নম্বর দিন থেকে দেখে নিন। অপারেটিং সিস্টেম এবার সেই তথ্যগুলো পড়বে টেপ থেকে। প্রোগ্রাম চলবে। এরপর, সব শেষ হলে, শেষ হওয়ার ঘোষণা, মানে এন্ড-কার্ড। এরপর আবার ঘুরে আসবে আর একটা কাজ, জব-কার্ড ২। আবার এই রকম একটা সিরিজ চলবে, সেটা হয়তো আবার ফোর্ট্রানে হবে না, হবে অ্যাসেম্বলি ভাষায়। তার মানে তার জন্যে আবার আলাদা রকমে কাজ করতে হবে অপারেটিং সিস্টেমকে।

দ্বিতীয় জেনারেশনের এই গাবদা কম্পিউটারগুলো মূলত ব্যবহার হত বড় বড় বিজ্ঞান গবেষণার আর ইঞ্জিনিয়ারিং-এর কাজে। মানে, মূলত বিশালাকার সব গাণিতিক হিশেব। যেমন পদার্থবিদ্যার বা আর্কিটেকচারের গাদা গাদা পার্শিয়াল ডিফারেনশিয়াল ইকুয়েশন সমাধান। এর বেশির ভাগ কাজটাই হত ফোর্ট্রানে বা অ্যাসেম্বলি ল্যাংগুয়েজে। এবং অপারেটিং সিস্টেমের বংশলতিকার ওই আদিপুরুষটি বেশিরভাগ ক্ষেত্রেই ছিল এফএমএস (FMS — **F**ortran-**M**onitoring-**S**ystem) বা আইবিএম-এর আইবিসিস। আমরা এখানে ফোর্ট্রান-মনিটরিং-সিস্টেমের একটা কাজের ছক দেখালাম।

৫।। তৃতীয় জেনারেশন কম্পিউটার — ১৯৬৫-১৯৮০

ততদিনে কম্পিউটার তৈরির দুটো টাইপ প্রায় আলাদা হয়ে এসেছে। একটা মেশিন বড় আকারের, বড় বড় আঙ্কিক হিশেব করার — মূল মনোযোগটা এখানে ওয়ার্ড বা শব্দের উপর। ওয়ার্ড মানে স্মৃতি বা মেমরির একক। এরা বড় বড় হিশেব এবং হিশেবনির্ভর কাজ করবে। ব্যবহার হবে বিজ্ঞান আর ইঞ্জিনিয়ারিং-এর কাজে। একটু আগে যেমন আইবিএম ৭০৯৪ দেখলাম আমরা। আর অন্য কম্পিউটারটা হল অপেক্ষাকৃত ছোট আকারের কমার্শিয়াল কম্পিউটার — মূলত ব্যবহার হবে ইনপুট তথ্যকে এবং আদেশকে টেপে ঢোকানো, আর আউটপুট তথ্যকে টেপ থেকে বার করে আনার কাজে। যেমন, একটু আগের আইবিএম ১৪০১। মূল মনোযোগটা এখানে ওয়ার্ড বা শব্দ নয়, ক্যারেকটার বা অক্ষর। শব্দ বা ধারণা নয়, শুধু অজস্র অক্ষরের পর অক্ষরের সমাহারে রাশি রাশি তথ্য। মূলত টেপে তথ্য তোলা এবং টেপ থেকে তথ্য পড়ার কাজে, এবং টেপে করে আসা তথ্য প্রিন্টারে প্রিন্ট নেওয়ার কাজে ব্যাপকভাবে এদের ব্যবহার করত অনেক ব্যাংক এবং বীমা কোম্পানি।

প্রযুক্তিবদলের সঙ্গে সঙ্গে দামও কিছু কমছিল, আর কম্পিউটার দিয়ে কাজ করানোর প্রয়োজনও বাড়ছিল। সরকার এবং বিশ্ববিদ্যালয় এবং মূলত ব্যবসা-প্রতিষ্ঠানগুলো আরো বেশি বেশি করে মেশিন কিনছিল। কিন্তু তাদের প্রয়োজনগুলো তো আর কম্পিউটারের ধরনের সঙ্গে মিলিয়ে তৈরি হয়নি। প্রায়ই তাদের, একইসঙ্গে, দুরকম মেশিনের দুরকম কাজেরই দরকার পড়ছিল। বা, কিছুদিন ধরে চলতে থাকা এক ধরনের প্রয়োজন প্রায়ই বদলে যাচ্ছিল অন্য ধরনের প্রয়োজনে। তাই, এই দু টাইপের কাজ একই মেশিনে করতে পারার চাহিদা ক্রমে বাড়ছিল। আইবিএম আনল সিস্টেম/৩৬০ বলে একটা যন্ত্র। যা আগের বড় কম্পিউটারগুলোর সমান ক্ষমতার, এবং আকারে

অনেক ছোট। আর সবচেয়ে বড় কথা, এরা শুধু হার্ডওয়ার স্তরের অ্যাসেম্বলি কোড দিয়েই চলে না, সফটওয়ারও বোঝে। আর যেহেতু এই সিরিজের সবগুলো মেশিনেরই গঠন এক, একটা কম্পিউটারের প্রোগ্রাম নির্দ্বিধায় অন্য কম্পিউটারে চালানো যায়। ওয়ার্ডনির্ভর বিজ্ঞানের কাজ আর ক্যারেকটারনির্ভর ব্যবসায়িক তথ্যের কাজ — দুই-ই করতে পারে এই সিস্টেম/৩৬০।

এই আইবিএম সিস্টেম/৩৬০ মেইনফ্রেম সিস্টেমে ব্যবহারের জন্যে এসেছিল জেসিএল (JCL — **J**ob-**C**ontrol-**L**anguage), একটা কমান্ড ল্যাংগুয়েজ যা দিয়ে কোনো প্রয়োগমূলক অ্যাপ্লিকেশন প্রোগ্রাম চালু করা যেত এবং ওই প্রোগ্রামের আকার, কত সময় ধরে চলবে, এবং চলতে গিয়ে প্রোগ্রামটা কী কী ফাইল ব্যবহার করবে — এইসবই ঠিক করে দেওয়া যেত। এই কমান্ড ল্যাংগুয়েজ হল কমান্ড ইন্টারপ্রিটারের কাজের একটা অংশ। ব্যাশ শেলের প্রসঙ্গে কমান্ড ইন্টারপ্রিটারের কথা পেয়েছি আমরা, আরো পাব।

সিস্টেম/৩৬০ মেশিনেই এল আইসি বা ইন্টিগ্রেটেড সার্কিট। দ্বিতীয় জেনারেশন কম্পিউটারগুলোর সঙ্গে দাম এবং ক্ষমতার লেভেলে সবচেয়ে বড় তফাত গড়ে দিল এই আইসি। জনপ্রিয় এই সিস্টেম/৩৬০ অনেক বিক্রি করেছে আইবিএম, পরে এই সিরিজে অন্য মডেলও এনেছে। যারা পরস্পর কম্পেটিবল বা স্থানান্তরযোগ্য, মানে এমন কম্পিউটার যাদের একের কাজ বা তথ্য বা প্রোগ্রাম অন্যের কাছে গ্রহণযোগ্য। কম্পেটিবল হওয়ার ধারণাটাও কম্পিউটার নির্মাতাদের কাছে জনপ্রিয় করেছে আইবিএম। সিস্টেম/৩৬০-এর কিছু উত্তরপুরুষ আজো ব্যবহৃত হচ্ছে, বিশাল বিশাল ডেটাবেস নিয়ন্ত্রণের কাজে, যেমন টিকিট রিজার্ভেশন বা ওই গোছের, বা ইন্টারনেটের সার্ভার হিশেবেও। পরবর্তী বছরগুলোয় আইবিএম এই সিস্টেম/৩৬০ সিরিজে আরো অনেক মডেল এনেছিল — ৩৭০, ৪৩০০, ৩০৮০, ৩০৯০ ইত্যাদি।

কিন্তু এই পরস্পর কম্পেটিবল কম্পিউটার পরিবারের ধারণাটায় সুবিধে যেমন অসুবিধেও আছে। নীতিটা এই যে, অপারেটিং সিস্টেম সমেত প্রতিটি সফটওয়ার এই পরিবারের প্রতিটি মেশিনে চলবে। তার মধ্যে কার্ড ফোটানোর দুধেভাতে কম্পিউটারও আছে, আছে আবহাওয়া পূর্বাভাসের মৈনাক মেশিনও। আর কুচো থেকে মৈনাকে তফাত শুধু আকারের নয়, তফাত তাদের কাজে, কাজে ব্যবহৃত তথ্যে, এবং সবচেয়ে বড় তফাত পার্শ্বীয় যন্ত্রপাতিতে মানে পেরিফেরাল ডিভাইসে। তার মানে একটা বিরাট সংখ্যক পেরিফেরালের প্রত্যেকটার সঙ্গে ব্যবহার করতে পারার ক্ষমতা এই পরিবারের প্রত্যেকটা কম্পিউটারেই ভরে দিতে হবে। তার মানে, দু নম্বর দিন থেকে মনে করুন — নানা ধরনের ড্রাইভার। ব্যবসা জগতের পেরিফেরাল থেকে বিজ্ঞান জগতের পেরিফেরাল।

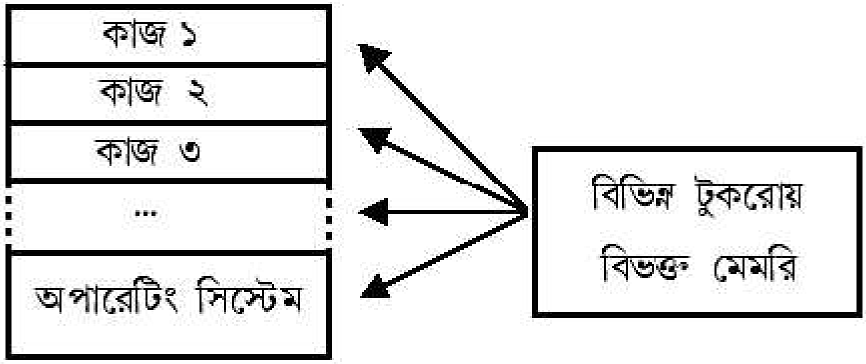
এই আলপিন টু এলিফ্যান্ট প্রতিটি মালের প্রতিটি চাহিদার সঙ্গেই চলনসই কোনো একটিমাত্র সফটওয়ার তৈরি করা আইবিএম কেন, কারো পক্ষেই সম্ভব ছিল না। এর ফলে যা হবার তাই হল। একটা বেধড়ক বদখত রকমের বড় সাইজের জটিল বহুমুখী অপারেটিং সিস্টেম। দ্বিতীয় জেনারেশন কম্পিউটারের ওই এফএমএস বা আইবিএমসিসের অন্তত তিনগুণ সাইজের। কোটি কোটি লাইনের অ্যাসেম্বলি কোড — যার পরতে পরতে অর্বুদ অর্বুদ বাগ। বাগ মানে তো বলেছি, প্রোগ্রামের ভুল, মানে খিঁচ, মানে মেশিন ঝুলে গেল, যা করার কথা তা না-করে আর সমস্ত কিছু

করারই অন্য নাম বাগ। এই পুরোনো দেড় হাজার বাগ শায়েস্তা করতে লেখা হল নতুন করে সোয়া দুলক্ষ লাইন অ্যাসেম্বলি কোড, তার লাইনে লাইনে জন্ম নিল আরো পৌনে দুহাজার বাগ। পোকামাকড় সংক্রান্ত এই বীভৎসাবিলাস সত্ত্বেও তৃতীয় জেনারেশন কম্পিউটার, সিস্টেম/৩৬০ এবং তাদের ওই ঘটোৎকচ অপারেটিং সিস্টেম পয়দা করে গেল একটা চমৎকার নতুন ধারণা — মাল্টিপ্রোগ্রামিং।

৬।। মাল্টিপ্রোগ্রামিং

সিস্টেম/৩৬০ এবং ওই জাতীয় নতুন কাঠামোগুলোয় আইসি বা ইন্টিগ্রেটেড সার্কিট ছাড়া আর একটা যে অভিনব বিষয় আমরা পেলাম সেটা হল মাল্টিপ্রোগ্রামিং। এই সিস্টেমগুলো এদের ভিতরকার ওই অযুত পোকামাকড় নিয়েই খদ্দেরদের চাহিদা বেশ ভালই মেটাচ্ছিল। এই মেটানোর প্রক্রিয়ায় যে সমস্ত নতুন নতুন ধারণাকে বাজারে এনেছিল তৃতীয় জেনারেশন, তার মধ্যে সবচেয়ে গুরুত্বপূর্ণ এই মাল্টিপ্রোগ্রামিং।

এর আগেকার বড় বড় সিস্টেমগুলোয় চলমান একটা কাজ যখন একটু থামত, নতুন টেপ গোঁজার ব্রেকে, বা, কোনো মহাভারত মাপের ইনপুট-আউটপুট জব শেষ হওয়ার অবসরে, হয়তো বহু পাতার একটা বড় প্রিন্ট চলছে, সিপিইউ বা সেন্ট্রাল প্রসেসিং ইউনিট একদম চুপচাপ বসে থাকত — যতক্ষণ না টেপটা গোঁজা হয় বা প্রিন্টটা শেষ হয়, আবার কাজ শুরু করার সুযোগ আসে। বহু বহু লাইনব্যাপী কোনো খেপচুরিয়াস বৈজ্ঞানিক হিশেবের খুব সিপিইউ-নিবিড় কোনো কাজ চলাকালীন এই ধরনের আইও ইন্টারমিশন হয়ত কমই হত, মানে সিপিইউ-র কাজের সময়ের তুলনায় বসে থাকার সময়ের হার খুব কমই থাকত। কিন্তু অন্য অন্য কাজে, কোনো বাণিজ্যিক ডেটা প্রসেসিং বা ওই ধরনের কিছু, এই ইনপুট-আউটপুট অপেক্ষার প্রহর বেড়ে বেড়ে পৌঁছে যেত আশি বা নব্বই শতাংশে। সিপিইউ-র সক্রিয় থাকার সময়ের হার কখনো কখনো নেমে গিয়ে মোট সময়ের দশ শতাংশেও দাঁড়াত। তার মানে, অন থাকার মোট সময়ের দশ শতাংশ মাত্র কম্পিউটারের মূল শক্তিটা ব্যবহৃত হচ্ছে। এ তো ঠিক না। এত দাম দিয়ে কেনা মেশিন এতক্ষণ কেন বসে থাকবে?



এর সমাধান আকারে এল একটা নতুন পদ্ধতি। স্মৃতি বা মেমরিটাকে পিসপিস করে ফেলা ক্ষুদ্রতর কিছু টুকরোয়। দুই নম্বর দিনে আমাদের মাল্টিপ্লেক্সিং নিয়ে আলোচনাটা মনে করুন। স্মৃতি বা মেমরির এই প্রতিটি পিসের হাতে এখন এক একটা ছোট ছোট কাজ ধরিয়ে দেওয়া হবে। কাজ নম্বর ১ যখন আইও ব্রেক মেটার অপেক্ষা করছে জিভ পুড়িয়ে ফেলা গরম চায়ের কাপে চুমুক দিতে দিতে, কাজ নম্বর ২ সঙ্গোপনে সিপিইউর সঙ্গে ডাবল-টাইম ডেটিং শুরু করে দিয়েছে — ইংরিজিতে যাকে বলে পরকিয়া। এবং এই গোটা সময়টা ধরে এই শতধা বিদীর্ণ মেমরির একটা বড় খণ্ড যত্নআত্তি করে যাচ্ছে প্রভুর প্রভু অপারেটিং সিস্টেমের। কখন সে কোন প্রসেসকে কোন আদেশ পাঠাবে বা কোন প্রসেসের থেকে আসা কোন আবেদনকে মঞ্জুর করবে — এই তথ্য এদিক থেকে ওদিক এবং ওদিক থেকে এদিক বহন প্রতিবহন করবার জন্যে। এই পুরো ব্যাপারটার বিউটি এইখানে যে এতে করে মহার্ঘ সিপিইউ-সুন্দরীকে প্রায় গোটা সময়টাই প্রায় একশো শতাংশতেই ব্যবহার করা যাচ্ছে। শুধু, সিপিইউকে মাল্টিপ্রোগ্রাম পদ্ধতিতে এই বহুগামিনী করে তোলার শর্ত একটাই। তখনকার সময়ের তুলনায় একটু আলাদা রকমের একটা হার্ডওয়ার যা স্মৃতির প্রতিটি খণ্ডকেই তার নিজস্ব একটা মহল বানিয়ে দেবে, এবং তার সঠিক প্রতিরক্ষার ব্যবস্থা করবে যাতে সে অন্যদের সঙ্গে ঘেঁটে না-যায়। স্মৃতির একটা টুকরোয় ভরে রাখা তথ্য যাতে অন্য টুকরোর মধ্যে উপচে না যায়। কোনো একটা

জব যাতে অন্য কারো ঘরে ছিঁচকেমি না করতে পারে। এই ছিঁচকেমির টেকনিকাল নাম নাম সেগমেন্টেশন ফল্ট ইত্যাদি — যার কথায় আমরা পরে আসব। ইন্টিগ্রেটেড সার্কিটের উদ্ভাবন এই তৃতীয় জেনারেশনেই, আগেই বলেছি, তার সঙ্গে সঙ্গে এই নতুন ধরনের হার্ডওয়ারের চাহিদাও মেটাতে হয়েছিল ওই সময়ের ইলেকট্রনিক ইন্ডাস্ট্রিকে।

তৃতীয় জেনারেশন কম্পিউটারের আর একটা বড় অবদান ছিল পাঞ্চড কার্ড পড়ার অযথা সময় নষ্টটাকে বন্ধ করা। যেই মুহূর্তে কম্পিউটার কক্ষে এসে ফুটোময় কার্ডমালা আবির্ভূত হল, সেই মুহূর্তেই তাদের পড়ে ফেলার ব্যবস্থা। এতে একটা কাজ শেষ করেই মেশিন পরের কাজে হাত দিতে পারে। সিপিইউ যখন একটা কাজ করায় রত, সেই সময়টা জুড়েই পরের কাজের প্রয়োজনীয় কার্ডমালা পড়ে রাখা হচ্ছে, মাল্টিপ্রোগ্রামিং ব্যবহার করে। এবার, চালু কাজটা নামানো মাত্রই সিপিইউর গরম তাওয়ায় পরের কাজ চড়িয়ে দেওয়া হচ্ছে। এর অন্য নাম স্পুলিং। একসঙ্গে অনেকগুলো পেরিফেরালকে ব্যস্ত রাখার ব্যবস্থা (SPOOL — **S**imultaneous-**P**eripheral-**O**peration-**O**n-**L**ine)। স্পুল বাজারে ল্যান্ড করার সঙ্গে সঙ্গে উঠে গেল টেপ কাঁধে অপারেটরদের ওই অন্তহীন হাঁটাহাঁটি।

মাল্টিপল প্রোগ্রামিং, স্পুল ইত্যাদির দৌলতে অনেক বড় বড় বৈজ্ঞানিক হিশেবনিকেশ এবার করা যাচ্ছিল। তাও, এর পরেও, তৃতীয় জেনারেশন-ও শেষ অব্দি এক ধরনের উন্নততর ব্যাচ সিস্টেম। একদিকে ঝামেলা যেমন কমিয়েছিল, অন্যদিকে, টেপ জমা দেওয়া আর তার ফলাফল পাওয়ার মধ্যে বেশ কয়েক ঘন্টার ব্যবধান তৈরি হয়ে গেছিল, যা আগে ছিলনা। প্রোগ্রামে একটা ভুল সেমিকোলন, কম্পাইলেশন খতম, কাজ বন্ধ। কিন্তু ততক্ষণে একটা গোটা দিন কাবার। সেমিকোলন শুধরে আবার পরের দিন। অনেক প্রোগ্রামচিই তাই চাইছিলেন সম্পূর্ণ তার নিজের করে কম্পিউটারের সঙ্গে একান্তে কিছুক্ষণ। এর থেকে ক্রমে গজিয়ে উঠছিল টাইম-শেয়ারিং-এর চাহিদা — আর একটু আলাদা অবয়বের মাল্টিপ্রোগ্রামিং। যখন মূল সিপিইউ একই সঙ্গে একাধিক টার্মিনালে যুক্ত থাকবে। এবং এক একটা টার্মিনাল থেকে আসবে এক এক সেট কাজ। এই সমস্ত চাহিদাগুলোই বাজারে পৌঁছে তৈরি করছিল চতুর্থ জেনারেশন কম্পিউটারের সম্ভাবনা।

৭।। সময়ভাগের ব্যবস্থা — সিটিএসএস

তৃতীয় জেনারেশন কম্পিউটার ব্যবহার করতে করতে নতুন ধরনের চাহিদা তৈরি হচ্ছিল, মূলত, প্রোগ্রাম লিখে ফেলার পর থেকে প্রোগ্রাম চালিয়ে হাতে ফলাফল পাওয়ার মধ্যে এই প্রতীক্ষার প্রহরটাকে কমিয়ে আনার জন্যে, একান্তে শুধু আমি আর কম্পিউটার — প্রোগ্রামলিখিয়েদের এই বাসনা থেকে। তাই তৃতীয় জেনারেশনেই একটু একটু করে আসছিল টাইম-শেয়ারিং বা সময়ভাগ। তৃতীয় জেনারশন চলতে চলতেই, মাল্টি-প্রোগ্রামিং পদ্ধতিটা একটু একটু করে অন্য চেহারা নিচ্ছিল। এই প্রক্রিয়ায় কাজ ১ কাজ ২ কাজ ৩ ... ইত্যাদিতে গোটা রসদটাকে বরাদ্দ করে ভেঙে ভেঙে দেওয়া হচ্ছিল ঠিকই, কিন্তু সেই কাজগুলো সবই কম্পিউটারে গোঁজা হচ্ছিল একই টেলিটার্মিনাল থেকে, গ্নু-লিনাক্স-এ যাকে আমরা সংক্ষেপে 'tty' বলে ডাকি। এই টেলিটার্মিনাল শব্দটা আর একটা লুপ্ত মেটাফর। সেই সময় সেই প্রকৌশল সবই বদলে গেছে, শুধু বদলে যাওয়া ভাষায় তার পদচিহ্ন রেখে গেছে — আমরা আজো ব্যবহার করে চলেছি, নিজের পিসিতে লাগানো কনসোলকেও ডাকছি টেলিটার্মিনাল বলে, টাকা হিশেবে কড়ির ব্যবহার ভুলে গিয়েও যেমন তেল মাখতে গেলেই কড়ি ফেলি। যাকগে, এই জাতের সময়ভাগ ব্যবস্থাটাকে একবার ভাবার চেষ্টা করুন। কুড়ি জন প্রোগ্রামচি সমভিব্যাহারে যুক্ত সিপিইউ-র সঙ্গে, কুড়ি পিস টেলিটার্মিনালে তারা বসে আছে, তাদের কাজ পাঠাচ্ছে সিপিইউর কাছে, সিপিইউ তাদের যথারীতি ঠেলে দিচ্ছে বিভিন্ন ধরনের রসদের বরাদ্দ কামরায়, স্মৃতির হোক, হার্ডডিস্কের, বা এমনকি আইও বা ইনপুট-আউটপুটের লাইনে। দু নম্বর দিনে আমাদের আলোচনাটা মনে করুন, যেখানে আমরা স্পেস এবং টাইম, ভূমি এবং কাল, এই দুই রকমে মাল্টিপ্লেক্সিং নিয়ে কথা বলেছিলাম। কখনো পরপর খদ্দের আসছে লাইন বেয়ে, ঘুরে ঘুরে, ডিপার্টমেন্টাল স্টোরে যেমন হয় — কালগত মাল্টিপ্লেক্সিং। আর কখনো, হোটেলের রুমের মত, এক এক জন খদ্দেরকে এক এক সঙ্গে ছেড়ে দিচ্ছি এক একটা আস্ত ঘর, প্রথম জন ঘর ছেড়ে দিলে পরের জন ঢুকছে — ভূমিগত মাল্টিপ্লেক্সিং।

এবার, এটা স্ট্যাটিস্টিক্সের একটা সোনালি সূত্র যে, কুড়িখানা লোক যদি একই সঙ্গে বসে থাকে তাদের কাজ নিয়ে, তাদের মধ্যে সতেরোজন অবধারিত বসে বসে মস্তি মারছে, হয়তো গেমস খেলছে, হয়তো কফি খাচ্ছে, বা নিট আড্ডা দিচ্ছে পাশের প্রোগ্রামচির সঙ্গে। তার মানে একটা প্রদত্ত মুহূর্তে সিপিইউ-কে আদতে ডিল করতে হচ্ছে

তিনপিস কাজ। আর প্রোগ্রামচিদের সবাই যে নতুন প্রোগ্রাম কম্পাইল করছে তা নয়, অনেকেই পোকা বাছছে পুরোনো প্রোগ্রামের। ডিবাগিং-এর কমান্ডগুলো সচরাচর হয় খুব ছোট ছোট। কম্পিউটারের সিপিইউ খুব দ্রুত এদের চাহিদা মিটিয়ে দিয়ে অন্য কাজে ফিরে যেতে পারে। আর এর সঙ্গে, ব্যাকগ্রাউন্ডে, অন্য কাজগুলোর ফাঁকে দম ফেলার ফুরসতের সময়খণ্ডে সিপিইউ করে চলতে পারে কয়েকটি জরুরি ব্যাচবদ্ধ কাজ। আবার সরাসরি নিবিড় কাজের চাপ চলে আসা মাত্রই ব্যাচবদ্ধ কাজ স্থগিত থাকছে। এই নিষ্পলক দাসত্ব স্পার্টাকাসও দেখেনি।

প্রথম সত্যিকারের কেজো সময়ভাগ ব্যবস্থা চালু হয় ম্যাসাচুসেটস ইনস্টিটিউট অফ টেকনলজিতে, ১৯৬২-তে, সিটিএসএস (CTSS — **C**ompatible-**T**ime-**S**haring-**S**ystem)। বিশেষ রকমে বদলে নেওয়া আইবিএম ৭০৯৪ সিস্টেম ব্যবহার করা হয়েছিল এই সিটিএসএস-এর জন্যে। কিন্তু তখনো টাইমশেয়ারিং-এর চূড়ান্ত জনপ্রিয়তার প্রহর আসতে দেরি আছে, তার কারণ, প্রতিটি কাজকে তার নিজের রসদের মহলে যথোচিত মর্যাদার সঙ্গে প্রতিরক্ষা আর প্রাইভেসি দেওয়ার মত হার্ডওয়ার তখনো বাজারে ল্যান্ড করেনি। নতুন ধরনের হার্ডওয়ার বানানো শুরু হওয়ার সঙ্গে সঙ্গে, তৃতীয় জেনারেশনের কম্পিউটারেই, সময়ভাগ ক্রমে সমূহ জনপ্রিয় এবং পরিব্যাপ্ত হয়ে ওঠে । সিটিএসএস-এর সাফল্যের পর এবার এমআইটি, বেল ল্যাবরেটরি, জেনারেল ইলেকট্রিক সবাই লাইন করে লাগল একটা কম্পিউটার কুম্ভ নির্মাণের — যে একাই গোটা বুঁদির কেল্লা সামলে দেবে। তার শিরায় শিরায় সেঁটে দেওয়া হবে শ-শ হাজার হাজার ব্যবহারকারী, লোকপিছু এক পিস করে টেলিটার্মিনাল। ঠিক বিদ্যুৎব্যবস্থার মত, একটা কেন্দ্রীয় কম্পিউটার তার সীমাহীন অনন্ত ক্ষমতা নিয়ে খুশি করে চলবে গোটা জনগোষ্ঠীকে।

আমাদের আবহমান ঈশ্বরকেন্দ্রিক বা এসেন্সকেন্দ্রিক চিন্তাপ্রক্রিয়ার সঙ্গে চিন্তার এই গোটা মডেলটা এমন খাপে খাপে মিলে যায় যে আজো, সাইফির পর সাইফিতে, চোয়াল নামিয়ে গলার উচ্চারিত আদমের আপেলে ঠেকিয়ে, ‘আয়াম ইওর ওয়ার্স্ট নাইটমেয়ার’ বলতে বলতে, স্ট্যালোন তার ভিলেনব্যূহকে ফাইনাল চূড়ান্ত টার্মিনাল রকমের শায়েস্তা করে দেয় গোটা গ্রহ জুড়ে ছড়ানো কম্পিউটার ব্যবস্থার হৃদপিণ্ডস্বরূপ কেন্দ্রীয় কম্পিউটারকে চটকে দিয়ে। এখানে মার্কিন সমাজমননের কাম পুঁজি-ব্যবস্থার একমাবেদ্বিতীয়ম ঈশ্বর হয়ে ওঠার, সবাইকে প্রত্যেককে প্রশ্নহীন ভাবে নিয়ন্ত্রণ করার আদি ফ্যান্টাসিরও একটা ছায়া থাকে — তার সুপারকম্পিউটারের অগোচরে কোথাও একটা কেশও উৎপাটিত না-হওয়ার।

৮।। বহু নদীর মোহনা — মাল্টিক্স

বহু ব্যবহারকারীকে, মাল্টিপল ইউজারকে, একসঙ্গে কাজ করতে দেওয়ার এই নতুন কম্পিউটারের পরিকল্পিত ছকটা পরিচিত হল মাল্টিক্স বলে (MULTICS — **MULT**iplexed-**I**nformation-and-**C**omputing-**S**ervice)। গোটা বোস্টন শহরের প্রত্যেকটি ব্যবহারকারীর প্রত্যেকটি কাজ যে একাই সামলাবে। এই চিন্তার একটা বড় এলাকা জুড়ে ছিল সেই আমলের মেশিনের খরচকে প্রত্যেকটা ব্যবহারকারীর একটা সম্ভাব্য দূরত্বের মধ্যে নিয়ে আসা। একটা বড় মেশিন সবার ভাগে ভাগ হলে খরচ কমে যাচ্ছে, ইকনমি অফ স্কেল। কিন্তু শত চেষ্টাতেও এই বিজ্ঞানীদের কাছে সম্ভব ছিলনা ভবিষ্যতকে দেখা। এবং, একটা নাটকীয় ভাবে বদলে যাওয়া ভবিষ্যত। তাদের গোটা চিন্তাটার ভিত্তিতেই এই গলদটা রয়ে গেছিল। তখন কেউ যদি তাদের কাছে বলত যে সেদিনের সবচেয়ে বেশি শক্তির কম্পিউটারের চেয়েও বহু কোটি গুণ বেশি ক্ষমতার কম্পিউটার মাত্র তিরিশ বছরের মধ্যে সেদিনের দামের তুলনায় জলের দরে বিকোবে গোটা পৃথিবী জুড়ে — এর চেয়ে বড় সাইফি তাদের কাছে আর কিছু হতনা।

মাল্টিক্স-এর অভিজ্ঞতায় ভালো আর মন্দ দুই-ই ছিল। আজকে আমাদের কাছে অভাবনীয় লাগবে — ইন্টেল থ্রিএইটসিক্স মেশিনের চেয়ে সামান্য বেশি শক্তির একটা সিপিইউর সঙ্গে যুক্ত করা হয়েছিল কয়েকশো ব্যবহারকারীর টেলিটার্মিনালকে — যদিও তার ইনপুট/আউটপুট ক্ষমতা ছিল অনেকটা বেশি। নিজের সময়ের প্রকৌশলের সঙ্গে মাল্টিক্স-এর গরমিলটা অনেকটা চার্লস ব্যাবেজের অ্যানালিটিকাল ইঞ্জিনের মত। অবশ্য আজকের আমাদের কম্পিউটার কাজের সবচেয়ে আগ্রাসী রসদভুক মাল্টিমিডিয়া এবং জিইউআই তখন ছিলনা। জিইউআই বা গুই মানে গ্রাফিকাল ইউজার ইন্টারফেস। বলেছি আগেই, কিবোর্ডের বদলে ছবি আর ইঁদুর দিয়ে কাজ করা। কনসোলে কমান্ড-প্রম্পটে কমান্ড লিখে এন্টার মারার জায়গায় জাস্ট একবার ক্লিক করে ইঁদুরের পেট টিপছি। সঙ্গে সঙ্গে দেখছি চলৎচিত্র — ফাইলের ছবি উড়তে উড়তে একটা ফোল্ডারের ছবি থেকে আর একটা ফোল্ডারের ছবিতে গিয়ে

পড়ছে। এই চলৎ-চিত্র ভারি ভারি উঁচু দামে কেনা। গোপনে আমার মেশিনের যাবতীয় হার্ডওয়ার রসদের একটা সিংহভাগ নিরবচ্ছিন্নভাবে গিলে চলে এই গুই। এই গুই-এর ল্যাটাই ছিলনা মাল্টিক্স-এর আমলে। আর, আরো একটা বড় ব্যাপার এই যে, রসদের ওই টানাটানির ভিতর কাজ করতে গিয়ে তখনকার প্রোগ্রামচিরা অনেক দক্ষ ছোট সাইজের এবং অনেক কম রসদ-হ্যাংলা প্রোগ্রাম লিখতে জানতেন। যাই হোক, যত তৃতীয় বিশ্ব গোছের চাহিদা নিয়েই আসুক তারা, তাদের মোট চাহিদা মেটানো মাল্টিক্সকালীন হার্ডওয়ারের দ্বারা সম্ভব ছিলনা, এছাড়া ‘পিএল’ কম্পাইলার সংক্রান্ত কিছু সমস্যাও ছিল, তার কথায় আসছি — সব মিলিয়ে মাল্টিক্স লুপ্ত হল, কিন্তু তার পূজাবিধি লুপ্ত হলনা। মাল্টিক্স উত্তরাধিকার রয়েই গেল, নিজের সময়ের, রসদের, হার্ডওয়ারের তুলনায় বড্ড বেশি উচ্চাকাঙ্খী মাল্টিক্সের কাঙ্খিত হার্ডওয়ার যখন ক্রমে দেখা দিতে শুরু করল মর্তভূমিতে।

সামগ্রিক কম্পিউটার ভাবনায় কিছু দীর্ঘস্থায়ী গভীর মোচড় এনে দিল মাল্টিক্স, যাদের প্রায় গোটাটাই আমরা আজো বয়ে নিয়ে চলেছি। কিন্তু মাল্টিক্সকে ব্যবসায়িক সাফল্য করে তোলা গেলনা কিছুতেই। বেল ল্যাবরেটরি প্রোজেক্টটা ছেড়ে দিল, জেনেরাল ইলেকট্রিক তো কম্পিউটার-ব্যবসাই ছেড়ে দিল। কিন্তু এমআইটি প্রোজেক্টটা চালিয়ে গেল, এবং শেষ অব্দি মাল্টিক্স মেশিন খাড়াও করতে পারল। প্রায় আশিটা বড় কোম্পানি এবং বিশ্ববিদ্যালয়ে মাল্টিক্স ইনস্টল করা হয়েছিল। এই সব জায়গার ব্যবহারকারীরা মাল্টিক্স-এর প্রতি চূড়ান্তভাবে বিশ্বস্ত। জেনেরাল মোটরস, ফোর্ড, এবং নাসা তাদের মাল্টিক্স ব্যবহার বন্ধ করল এই সেদিন, নব্বই-এর দশকের শেষ দিকে — মাল্টিক্স ব্যবহার শুরু করার প্রায় তিরিশ বছর পরে।

ব্যবসায়িক সাফল্য না পেলেও মাল্টিক্স ধারণাটা এর পরবর্তী কম্পিউটার ডিজাইনগুলোকে এবং কম্পিউটার চিন্তাকে বদলে দিল। পরবর্তী কম্পিউটার প্রজন্মের উপর মাল্টিক্স-এর প্রভাব বিস্তারিত ভাবে নেটে আছে, নানা সাইটে, বিশেষ করে মাল্টিক্স-এর নিজের ‘www.multicians.org’ ওয়েবসাইটে। পরে আমরা এই নিয়ে আরো কথা বলব, গ্নু-লিনাক্স শিখতে চাইলে বলতেই হবে। ওপন-সোর্স ইউনিক্স মানে গ্নু-লিনাক্স তথা যে কোনো ইউনিক্সেই, যে কোনো ভৌত উপাদান বা ফিজিকাল ডিভাইসকেই দেখা হয় একটা ফাইল হিশেবে — ডিভাইসকে ফাইল হিশেবে দেখার এই ধারণাটা হল মাল্টিক্স-এর উত্তরাধিকার। হানিওয়েল বলে একটা প্রতিষ্ঠান, যারা জেনেরাল ইলেকট্রিকের কম্পিউটার ব্যবসার অংশটা কিনে নিয়েছিল, তারা এই মাল্টিক্স-এর ব্যবসায়িক সংস্করণও বাজারে ছাড়ে। তবে খুব একটা জনপ্রিয় হয়নি। মাল্টিক্স-এর আর একটা বড় অবদান কেন থম্পসন, যার কথায় আমরা এখুনি আসব, ইউনিক্স-এর প্রসঙ্গে। কেন থম্পসন তার প্রোগ্রামিং-এর কাজ শুরু করেছিলেন এই মাল্টিক্স দিয়েই। আগেই বলেছি, মাল্টিক্স-এর ব্যর্থতার আর একটা কারণ ছিল এর কুখ্যাত পিএল কম্পাইলার (PL/I — **P**rogramming-**L**anguage-I)। এই কম্পাইলার তৈরি করেছিল আইবিএম ১৯৬৪ থেকে ৬৯-এ। এতে ফোর্ট্রান কোবল, এবং অ্যালগল — তখনকার চালু এই তিনটে কম্পিউটার ভাষার মূল বৈশিষ্ট্যগুলোকে এক সঙ্গে নিয়ে আসা হয়েছিল, আর যোগ করা হয়েছিল ভুল-শোধরানোর ক্ষমতা এবং মাল্টিআস্কিং। কিন্তু এত বিতিকিচ্ছিরি রকমের জটিল হয়ে পড়ে এই পিএলআই, যে এতে কাজ করা প্রায় অসম্ভব ছিল। কেন থম্পসন এবং ডেনিস রিচির কৃতিত্বে এবার একটা মৌলিক বদল আসতে শুরু করে একদম এই প্রাথমিক জায়গা থেকেই। পরের সেকশনেই আসছি সেই কথায়।

৯।। মিনিকম্পিউটার, ইউনিক্স, সি

তৃতীয় জেনারেশন চলতে চলতেই মেশিনের বাজার নাটকীয় রকমে বদলে গেল। বাজারে বিপুল সংখ্যায় আসতে শুরু করল মিনি-কম্পিউটার। বড় সিস্টেমগুলোর থেকে এদের রিসোর্স বা রসদ বেশ কম, কিন্তু দাম বড় মেশিনের কুড়ি ভাগের এক ভাগ। বৈজ্ঞানিক আঙ্কিক কাজ যাদের নয়, তাদের জন্যে গন-উৎপাদিত হতে শুরু করল ছোট মেশিন। শুরু ১৯৬১-তে, ডিইসি পিডিপি এক থেকে, চলল এগারো অব্দি। এই পিডিপি ১-এর ক্ষমতা ছিল ১৮-বিটের শব্দের চার হাজার শব্দ। এর দাম ছিল এক লাখ কুড়ি হাজার ডলার, মানে আজকের, দোসরা ডিসেম্বর ২০০৩-এর চালু ডলার বিনিময় ৪৬.৩০ টাকার হিশেবে, পঞ্চান্ন লক্ষ ছাপ্পান্ন হাজার টাকা। কিন্তু, সেটাও সেই মুহূর্তে একটা ৭০৯৪-এর দামের পাঁচ শতাংশেরও কম। তাই, পটলডাঙার মুখে দ্বারিকের তেলেভাজার মত বিক্রি হল। সংখ্যা চিবোনোর কাজ বাদ দিলে অন্য কাজে পিডিপি সিরিজ গতিতে ৭০৯৪-এর প্রায় সমান ছিল।

পিডিপি সিরিজ নির্মাতা ডিইসি (Digital Equipment Corporation) তৈরি হয়েছিল ১৯৫৭-য়। যাটে এসে এরা কম্পিউটার তৈরি করা শুরু করে, কিন্তু মেশিনের নাম দেয় কম্পিউটার নয়, পিডিপি (PDP — **P**rogrammed-**D**ata-**P**rocessor)। নামটা দেখুন, অনেকটা মুড়িকে মুড়ি না বলে চাল-ভাজা বলে ডাকার মত। এর একটা পরিস্কার বেওসায়িক কারণ ছিল। আসলে মেশিনের ইমেজটা বদলাতে চাইছিল ডিইসি। কম্পিউটার বলতেই তখন লোকে বুঝত বিশেষ ধরনের বিল্ডিং, বিশেষ ধাঁচের কর্মচারী, বিরাট একটা স্কেল। ডিইসি এই গোটাটাকে সরিয়ে দিতে

চাইছিল। আর ব্যবসা প্রতিষ্ঠানগুলোকে নিজের টার্গেট খদ্দের না-করে, তাক করেছিল বিজ্ঞান এবং ইলেকট্রনিক্স জগতের লোকেদের। এতে সফটওয়ার খরচও কমে গেছিল। এই লোকগুলো কম্পিউটারের কাজে অনেকটাই আত্মনির্ভর, নিজেদের পছন্দমত রকমে সফটওয়ার বানিয়ে নেবে, এবং বিক্রির পরে সহযোগিতাও অনেক কম করতে হবে। সাইবার বিজ্ঞানীরাও খুশি হয়েছিলেন চমৎকার খেলনা পেয়ে। নতুন নতুন প্রোগ্রাম লিখতে শুরু করেছিলেন। এই পিডিপি সিরিজের শেষটা ছিল পিডিপি ১১, যে মেশিনে বেল ল্যাবরেটরিতে কেন থম্পসন আর ডেনিস রিচির কাজ করার একটা ছবি দিয়েছি আমরা। যদিও, ওই ছবিটা আমি নিজেই জানিনা কপিরাইট আছে কিনা, বহু জায়গাতেই আছে, কিন্তু কম বাইটে ভালো রেজলিউশনের এটা পেয়েছি একটা জাপানি বা চিনা সাইটে। গুগলিতে ইমেজ সার্চ দিয়ে। কী ভাষা বুঝতে পারিনি, তাই ডিফল্টে ধরে নিতে অসুবিধে নেই যে, কপিরাইট নেই।

বেল ল্যাবরেটরিতে মাল্টিক্স সিস্টেম নিয়ে যে বৈজ্ঞানিকরা কাজ করছিলেন তাদের একজন ছিলেন কেন থম্পসন। ল্যাবরেটরিতে থম্পসন আপাত-দাবীদারহীন একটা মিনিকম্পিউটার হাতের কাছে পেয়ে গেলেন, যা সেই মুহূর্তে কেউ ব্যবহার করছে না, ওই পিডিপি সিরিজের পিডিপি ৭। একটা কাঁটছাঁট-করা, ছোটো সাইজের, একজন ইউজার বা ব্যবহারকারীর ব্যবহারের মাল্টিক্স অপারেটিং সিস্টেম কাঠামো বানিয়ে তোলার চেষ্টা শুরু করলেন থম্পসন এই মেশিনে। মাল্টিক্স লেখার ভাষা কিন্তু ওই পিএল-আই নয়, সরাসরি অ্যাসেম্বলি ল্যাংগুয়েজ। পিডিপি-সাত-এর কুচো

সাইজ আর সীমিত সামর্থ সত্ত্বেও, থম্পসন-এর এই চেষ্টা সত্যিই কাজ করেছিল। এবং এই কুচো সিস্টেমে দাঁড়িয়ে পরবর্তী বদল ও উন্নতির, ডিভেলপমেন্টের কাজটা করে যেতে পেরেছিলেন। থম্পসন কাজ করে চলাকালীন বেল ল্যাবরেটরির এক সহকর্মী ব্রায়ান কারনিঘান, মাল্টিক্স-এর নামের ধবনির সঙ্গে মিলিয়ে, মৃদু ক্যাওড়ামি সহ, এর নাম দেন ইউনিক্স (UNIX — UNICS — **Un**iplexed-**I**nformation-and-**C**omputing-**S**ervice)। মজাটা ছিল ‘ইউনাক’ বা খোজা শব্দটায়। মাল্টিক্স থেকে অনেককিছু গুরুত্বপূর্ণ ছেঁটে ফেলে একে পাওয়া গেছিল, ঠিক যেমন করে পাওয়া যেত প্রাসাদের হারেমের খোজাদের।

কেন থম্পসন এবং ডেনিস রিচি কাজ করছেন বেল ল্যাবের পিডিপি এগারোয়

পিডিপি ৭-এ এই গোড়াপত্তনের কিছুদিন পর কাজটাকে নিয়ে আসা হয় অনেকটাই উন্নততর একটা পিডিপি ১১-য়। বেল ল্যাবে কেন থম্পসনের সহকর্মীদের মধ্যে থেকে ডেনিস রিচি থম্পসনের এই প্রোজেক্টে যোগ দেন। এরপর যোগ দেন তার সমস্ত সহকর্মীই। এই শেষ মেশিনটা ছিল তার সময়ের তুলনায় খুবই এগোনো। মেমরির সাইজ ছিল দুই মেগাবাইট। এবং একটু আগে যে প্রতিরক্ষামূলক মেমরি প্রোটেকশন হার্ডওয়ারের কথা বলছিলাম আমরা, টাইমশেয়ারিং-এর প্রসঙ্গে, সেটাও ছিল এতে, যাতে একই সঙ্গে একাধিক ব্যবহারকারী একে ব্যবহার করতে পারেন। কিন্তু ১৬-বিট মেশিন হওয়ায় একটা একক প্রসেস বা পদ্ধতিতে আদেশ বা ইন্সট্রাকশনের জন্যে ৬৪ কিলোবাইট আর তথ্য বা ডেটার জন্যে ৬৪ কেবির বেশি থাকতে পারত না, ভৌত মেমরি অনেক বেশি হওয়া সত্ত্বেও।

এবার এল কী ভাষায় এই অপারেটিং সিস্টেমটা লেখা হবে সেই সমস্যা। এক নম্বর দিনে আমরা বলেছি, অপারেটিং সিস্টেমটাও একটা প্রোগ্রাম, গোটা ব্যবস্থার সবচেয়ে রেলাবাজ প্রোগ্রাম। তার চলার উপর দাঁড়িয়ে চলতে পারে অন্য আর সব যাবতীয় প্রোগ্রাম। এবং সময়টা ভাবুন, তখন অপারেটিং সিস্টেমও বানিয়ে নিতে হয়, আজকের মত হাতে গরম তৈরি কারনেল পাওয়া যায়না, যা মোটামুটি যে কোনো হার্ডওয়ারেই চলে। তার মানে, একটা মেশিন ছেড়ে যদি আপনার কাজ আপনি অন্য মেশিনে নিয়ে যেতে চান, তাহলে সেই কাজের পিছনের অ্যাপ্লিকেশন প্রোগ্রামটা আপনাকে সেই নতুন মেশিনে চালাতে হবে। সেটা চালাতে গেলে তার আগে চালাতে হবে সকল প্রোগ্রামের নবাব প্রোগ্রাম কারনেল বা অপারেটিং সিস্টেমকে। কিন্তু পুরোনো মেশিনের অপারেটিং সিস্টেম তো নতুন মেশিনে চলবে না, তাদের হার্ডওয়ার আলাদা হতে পারে, আইও ডিভাইস আলাদা হতে পারে, সেগুলোর আলাদা ড্রাইভারকে মিলিয়ে নতুন অপারেটিং সিস্টেম গজিয়ে নিতে হবে নতুন মেশিনে, যাতে এবার তার উপর দাঁড়িয়ে আপনার কাজের অ্যাপ্লিকেশন প্রোগ্রামটা আপনি চালাতে পারেন। এটাই স্থানান্তর বা ট্রান্সপোর্টেবিলিটির সমস্যা, চালু টেকনিকাল শব্দে ‘পোর্ট করা’ বলে ডাকা হয়, পরে একাধিকবার আমরা আসব এই প্রসঙ্গে।

তদ্দিনে মেশিন থেকে মেশিনে প্রত্যেকবার নতুন করে অপারেটিং সিস্টেম লিখে লিখে প্রোগ্রাম ডিভলপাররা হাড়ে হাড়ে বুঝেছেন এমন একটা অপারেটিং সিস্টেমের মাহাত্ম্য যা সব মেশিনেই সমান পটীয়ান হবে। থম্পসন নিজেই বানিয়েছিলেন একটা কম্পিউটার ভাষা, তার নাম বি। এই বি নামটা এসেছিল এর আগের একটা ভাষা বিসিপিএল-এর উত্তরাধিকার থেকে। বিসিপিএল (BCPL — **B**asic-**C**ombined-**P**rogramming-**L**anguage) এসেছিল

সিপিএল থেকে, ঠিক ওই পিএল-আই-এর মতই আর একটা ভাষা যা কোনোদিনই তার প্রতিশ্রুতিমত কাজ করে উঠতে পারেনি। প্রথমে থম্পসন ইউনিক্স-কে এই বি-তে লেখার কথা ভেবেছিলেন। কিন্তু বি-র নিজস্ব কিছু সমস্যার জন্যে এই চেষ্টা সফল হলনা। তখন ডেনিস রিচি এই বি-র একটা বংশধর বানালেন, তার নাম আর কী হতে পারে — বি-এর পর, সি ছাড়া? আর লিখলেন এই সি-এর একটা ভালো কম্পাইলার। এরপর, কেন থম্পসন আর ডেনিস রিচি মিলে ইউনিক্স নামের অপারেটিং সিস্টেমটাকে সি-তে লিখে ফেললেন।

বিভিন্ন বিশ্ববিদ্যালয় থেকে বিজ্ঞানীরা এবার ইউনিক্স চাইতে লাগল বেল ল্যাবরেটরি তথা রিচি-থম্পসনের কাছে। মার্কিন সরকারের সেই সময়কার একটা আইন মোতাবেক, বেল ল্যাবরেটরির মূল কোম্পানি এটিঅ্যান্ডটি-র পক্ষে কম্পিউটার সংক্রান্ত কোনো ব্যবসায় নামা সম্ভব ছিলনা। তাই, নামমাত্র একটা টাকায় এই ইউনিক্স পৌঁছে গেল বিশ্ববিদ্যালয়গুলোয়। আর পিডিপি সিরিজের বিজ্ঞানীচাটা ব্যাপারটা তো বলেছি আগেই। বিজ্ঞানীরা তাদের মনের মত একটা সিস্টেম পেয়ে গেলেন, কম্পিউটারের ইতিহাসে সবচেয়ে সুদূরপ্রসারী বদলগুলোর একটা জোড়া বদল ঘটে গেল একই সাথে — ইউনিক্স আর সি। ইতিহাস আসলে এই ভাবেই গজায়, কতকগুলো পরপর ঘটতে থাকা সমাপতন মিলে ফুটে ওঠে একটা ধাঁচ, প্রায় অলক্ষে, প্রায় অকারণে। মানে গূঢ়তর কোনো হেতু ব্যতিরেকেই। পরে আমরা, জ্যান্ত ঘটনা মরে ইতিহাস হয়ে গেলে, পড়ি যখন, চমকাই, আরে তাইতো — এই ইউনিক্স-এর বেলাতেও এইরকম একটা সমাপতন খুব গুরুত্বপূর্ণ জায়গা নিয়েছিল। এই বিশ্ববিদ্যালয়গুলোর অ্যাকাডেমিক আবহাওয়ায় বেশিরভাগ কেসেই মেশিনটা ছিল পিডিপি ১১, যাতে বানানো ইউনিক্স। তাই অনেক বেশি অনাবিলতায়, প্রায় কোঁৎ না-পেড়েই ইউনিক্স চালানো যেত এই মেশিনগুলোয়। ইউনিক্স-এর পতাকা পৎপৎ নয়, দুমদাম করে উড়তে শুরু করল। আর সি-তে লেখা মানে এই ইউনিক্স এবার একটা মেশিন থেকে অন্য মেশিনে নিয়ে যাওয়া, যার অন্য নাম পোর্ট করা, ভারি সহজ হয়ে গেল। শুধু ওই সি কম্পাইলারটুকুই লিখতে হবে নতুন মেশিনের মত করে, তারপরেই তাতে সি চালানো যাবে। আর সি চালাতে পারা মানেই তাতে ইউনিক্স চালাতে পারা।

ইউনিক্সের পরবর্তী প্রজন্মের বিভিন্ন বাণিজ্যিক ক্লোন যারা পোজিক্স মেনে তৈরি

নাম	নির্মাতা	মন্তব্য
জেনিক্স (Xenix)	মাইক্রোসফট	এখন অবলুপ্ত
বিএসডিআই (BSDi), বিএসডি অপারেটিং সিস্টেম (BSD/OS), ফ্রি-বিএসডি (FreeBSD)	বার্কলে সফটওয়ার ডিজাইন	বার্কলে সফটওয়ার ডিস্ট্রিবিউশন যারা করেছিলেন তাদেরই বাণিজ্যিক প্রয়াস
সান অপারেটিং সিস্টেম (SunOS)	সান মাইক্রোসিস্টেম	পরে সোলারিস-এর মধ্যেই ঢুকে যায়
সোলারিস (Solaris)	সান মাইক্রোসিস্টেম	পিসিতে চালানোর সংস্করণও আছে
এইচপি-ইউএক্স (HP-UX)	হিউলেট প্যাকার্ড	
আইক্স (AIX)	আইবিএম	
আলট্রিক্স (Ultrix)	ডিইসি	অবলুপ্ত, পরে ডিজিটাল ইউনিক্স এর জায়গা নেয়
ডিজিটাল ইউনিক্স (Digital UNIX)	ডিইসি	
আইরিক্স (IRIX)	সিলিকন গ্রফিক্স	
স্কো ওপন সার্ভার (SCO Open Server)	সান্টা ক্রুজ অপারেশন	পরে স্কো-ইউনিক্সওয়ার এর জায়গা নেয়
স্কো ইউনিক্সওয়ার (SCO UnixWare)	সান্টা ক্রুজ অপারেশন	পিসিতে চালানো যায়

তথ্যসূত্র — সুমিতাভ দাসের 'ইউনিক্স দি আলটিমেট গাইড'

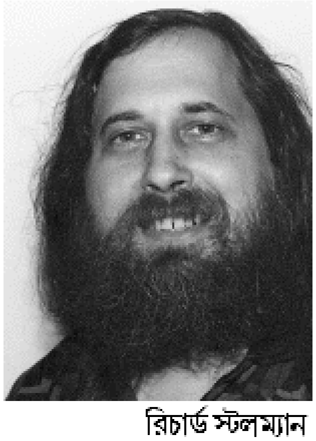
১০।। ইউনিক্স, মিনিক্স, লিনাক্স

আমরা এক নম্বর দিনে বলেছি, উইন্ডোজ অপারেটিং সিস্টেমের ক্ষেত্রে সোর্স-কোড উন্মুক্ত নয়। এর একটা আন্দাজ আমরা ইতিমধ্যেই পেয়েছি, তাও একটু ঝালিয়ে নিই এবার। কম্পিউটারে যাদের নিয়ে আমরা কাজ করি সেই অ্যাপ্লিকেশনগুলো প্রোগ্রাম, আবার তাদের কাজের ভিত্তি মানে অপারেটিং সিস্টেমও একটা প্রোগ্রাম। এই প্রোগ্রামগুলো সরাসরি বানানো যায়না। প্রোগ্রামটাকে বানাতে গিয়ে, প্রথমে লেখা হয় মানুষের বোধ্য রকমে, হাই

লেভেল কোনো কমপিউটার ভাষায়, যখন তার নাম কোড। ইউনিক্সের ক্ষেত্রে কোডের ভাষাটা প্রায় গোটাটাই সি। এই সি-তে বা অন্য ভাষায় লেখা কোডটা তারপর কম্পাইল করা হয়, মানে, কম্পিউটারের বোধ্য আদেশমালায় রূপান্তরিত করা হয় — লো লেভেল ভাষায়। লেভেলটা সত্যিই ভারি লো, আগেই যেমন বলেছি, কম্পিউটার তো সত্যিই বেশ কম বোঝে, সাধে আমরা গাধা বলে ডেকেছিলাম। তখন সেই কম্পাইলড কোডটা হয়ে দাঁড়ায় একটা প্রোগ্রাম। এই কম্পিউটার-বোধ্য আদেশমালায় বা প্রোগ্রামে রূপান্তরিত বা কম্পাইলড হওয়ার পর মূল মানুষবোধ্য কোডটা এবার একটা নতুন আকার পায়, এমন ফাইল যা নিজে নিজে চলে। ইউনিক্স বা সোর্স-উন্মুক্ত গ্নু-লিনাক্স জগতে এই ফাইলগুলোকে ডাকা হয় বাইনারি ফাইল বলে, আর উইন্ডোজ জগতে এক্সিকিউটেবল ফাইল। সেটা ইউনিক্স জগতের বাইনারি হোক, বা উইন্ডোজ জগতের এক্সিকিউটেবল, তাদের প্রত্যেকেরই তাহলে অন্য একটা আকার থাকে — একটা সোর্স কোড, মানুষবোধ্য মূল আকার, যাকে কম্পাইল করে এই চালনীয় আকারটা আসে।

আমরা যখন মাইক্রোসফট উইন্ডোজ কিনি, বা মাইক্রোসফটের কোনো প্রায়োগিক প্রোগ্রাম, এমএস-ওয়ার্ড জাতীয়, মাইক্রোসফট আমাদের ওই কম্পাইলড চালনীয় আকারটা বিক্রি করে, কখনো তার মূল আকার বা সোর্স কোডটা জানতে দেয়না। সিডি-তে লেখা সেই কম্পাইলড আকারটা আমরা নিজেদের মেশিনে চালাই। মূল প্রোগ্রামটা আমাদের জানার বা দেখার সুযোগ থাকেনা। গ্নু-লিনাক্স জগতকে ওপন-সোর্স জগত বলা হয় কারণ সেখানে কম্পাইলড আকারটা যেমন পাওয়া যায়, মূল প্রোগ্রামটাও পাওয়া যায়, যাকে আমরা কম্পাইল করতে পারি নিজের মেশিনে, বা বদলাতে পারি, যদি দরকার পড়ে। অর্থাৎ যে একটা নতুন অপারেটিং সিস্টেম বা অ্যাপলিকেশন সফটওয়ার বানাতে চায় সে ওই উন্মুক্ত সোর্স কোড থেকে কাজ শুরু করতে পারে।

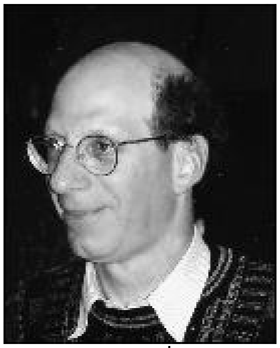
কিন্তু সোর্স কোড যদি গোপন বা ক্লোজড রাখা হয় সেক্ষেত্রে যে নতুন কিছু বানাতে চায় তাকে শুরু করতে হয় শূন্য থেকে, মানে মাইক্রোসফট যেখান থেকে শুরু করেছিল, তারপর মাইক্রোসফটের আবিষ্কৃত প্রতিটি পদক্ষেপ তাকে আবার নতুন করে আবিষ্কার করে নিতে হয়। আর এক ভাবে ভাবুন, মাইক্রোসফট কী? একটা পুঁজির নাম, যে কয়েকজন প্রতিভাবান প্রোগ্রামারকে পয়সার বিনিময়ে নিয়োগ করছে, তাদের কাজটাকে বাজারে নিয়ে আসছে নিজের ব্রান্ডনেম লাগিয়ে। ওই প্রোগ্রামারদের পরের প্রজন্ম তাদের পূর্বপুরুষের কাজের থেকে বঞ্চিত হল, তাদের ইতিমধ্যেই হেঁটে ফেলা পথ আবার করে হাঁটতে হবে। এটাকেই রিচার্ড স্টলম্যান বলেছেন রি-ইনভেন্টিং দি হুইলস, বা বারবার করে চাকা আবিষ্কার করে চলার চক্কর।



রিচার্ড স্টলম্যান

এই প্রসঙ্গটা এখানে আনতে হল, কারণ, প্রথম যখন থম্পসন এবং তার সঙ্গীরা মিলে ইউনিক্স বানান, সেটা ছিল ওপন সোর্স। যার কাজে লাগবে সেই সেটা পেয়েছে, পরে যখন বিভিন্ন ব্রান্ডনেম-এর ইউনিক্স বাজারে এল সেগুলো আর ওপন-সোর্স নয় (আরো বিশদ করে এটা জানব আমরা পাঁচ নম্বর দিন)। কেন থম্পসন-রা যেহেতু সোর্স কোডটা যে কারুর জন্যে উন্মুক্ত করে দিয়েছিলেন, তার থেকে বিভিন্ন প্রতিষ্ঠান বা বিশ্ববিদ্যালয় বা গবেষণা-সংস্থা তাদের নিজেদের মত ইউনিক্স সংস্করণ তৈরি করে নেয়, তাদের প্রয়োজনমাফিক সেই মূল কোডকে বদলিয়ে। কিন্তু এতে বিভিন্ন বিচিত্র রকমের ইউনিক্সের একটা খিচুড়ি তৈরি হয়ে যায় — তাদের একটা নিয়মনিষ্ঠ আকার দিতে গিয়ে পরে দুটো মূল ভার্শন আসে। একটা সিস্টেম-ভি, আর অন্যটা বিএসডি (বার্কলে সফটওয়ার ডিস্ট্রিবিউশন)।

এই ভার্শনগুলোর মধ্যে কিছু কিছু ছোটখাট তফাতও ছিল। একটা ইউনিক্সে তৈরি করা প্রোগ্রাম যাতে অন্য যে কোনো ইউনিক্সে চালানো যায় তার জন্যে তৈরি হল পোজিক্স (POSIX — **P**ortable-**O**perating-**S**ystem-**I**nterface) — প্রায়োগিক প্রোগ্রামগুলোর সঙ্গে কম্পিউটার হার্ডওয়ারের এবং অপারেটিং সিস্টেমের পারস্পরিক সম্পর্কের কিছু নির্দিষ্ট মাপকাঠি, কিছু মানদণ্ড তৈরি করে দেওয়া, যাতে প্রতিটি অপারেটিং সিস্টেমে আলাদা আলাদা করে সোর্স কোড তৈরি করতে না-হয়, একবার তৈরি করেই তা নানা অপারেটিং সিস্টেমে কম্পাইল করে নেওয়া যায়। পোজিক্স ছিল ইউনিক্স ঘরানার। পরে গ্নু-লিনাক্স জগতের এক্স/ওপন (X/Open Specifications) ব্যবস্থা পোজিক্স-এর উপরে নির্ভর করে তৈরি হয়েছিল। দুই নম্বর দিনে আমরা সিস্টেম-কলের কথা বলেছি, একটা সফটওয়ার যখন কোনো একটা রসদ ব্যবহার করতে দেওয়ার আবেদন জানায় অপারেটিং সিস্টেমের কাছে, সেই সিস্টেম-কলের কিছু নির্দিষ্ট ছক ঠিক করে দেয় পোজিক্স। নিজের ব্যবস্থাকে ইউনিক্স বলে ঘোষণা করতে গেলে যে ছক মানতে হবে। যাতে অন্যদের কাজের সঙ্গে আমার সিস্টেমের একটা সাযুয্য বা কম্পেটিবিলিটি আসে।



অ্যান্ড্রু ট্যানেনবম

১৯৮৭-তে ট্যানেনবম তার ছাত্রদের কম্পিউটার তত্ত্ব বোঝানোর প্রয়োজনে তৈরি করেন মিনিক্স বলে একটা অপারেটিং সিস্টেম। এটা এক অর্থে ছিল ইউনিক্স-এর একটা প্রতিরূপ — ক্লোন। ইউনিক্সের সঙ্গে মিনিক্সের ব্যবস্থা প্রায় হুবহু এক, শুধু আকারে অনেক ছোট। এবং মিনিক্স-ও এই পোজিক্স ব্যবস্থা মানে। গোটা সোর্স কোড সহ এই মিনিক্স অপারেটিং সিস্টেম এবং তার প্রয়োজনীয় পাঠ্য বিনামূল্যে নেটে পাওয়া যায়, 'www.cs.vu.nl' সাইটে। কিন্তু মিনিক্স-এর মূল ঝোঁক ছিল লেখাপড়া, বিদ্যাচর্চা। তাকে কেজো জায়গায় এনে অন্য অপারেটিং সিস্টেমের বিকল্প হিশেবে ব্যবহার করা প্রায় অসম্ভব। মিনিক্স-এর মডেলে, মিনিক্স ব্যবহার করে এই কাজটা শুরু করেন ফিনল্যান্ডের এক ছাত্র, লিনাস টরভাল্ডস। রিচার্ড স্টলম্যান তার বেশ কিছু বছর আগে থেকেই শুরু করে দিয়েছিলেন তার মুক্ত সফটওয়ার আন্দোলন, এবং তার সংগঠন আর ওয়েবসাইট, গ্নু (GNU — **G**NU's-**N**ot-**U**NIX)। এই আন্দোলনের অংশ হিশেবেই এসেছিল স্টলম্যানদের বানানো কম্পাইলার জিসিসি। আর মেক, অটোকনফ ইত্যাদি সফটওয়ার। যাকে ব্যবহার করেন লিনাস তার অপারেটিং সিস্টেমের কম্পাইলেশনে। এবং স্টলম্যানদের এই কাজগুলো সবই জিপিএল বা জিএনইউ পাবলিক লাইসেন্সের আওতায়। যা প্রচলিত কপিরাইট আইনের বিরোধী, কিন্তু তার মানে কপিরাইট ছেড়ে দেওয়া নয়, অন্য ধরনের একটা কপিরাইট।

লিনাস টরভাল্ডস

ধরুন আপনি একটা সফটওয়ার বানালেন। সেটা আপনি জিপিএলের আওতায় আনলেন। তার মানে এই নয় যে আপনি সেটা বিনামূল্যে দেবেন। আপনি সেটা বিক্রি করতেই পারেন। কিন্তু সেই সফটওয়ার যদি কেউ বিনামূল্যে দেয়, তাহলে আপনার কোনো বাধা দেওয়ার অধিকার থাকবে না। বা, সেই মূল সফটওয়ার ব্যবহার করার এবং বদলানোর অধিকার থাকবে যে কারুরই। কিন্তু যেহেতু সেটা একটা জিপিএল-এর আওতায় থাকা সফটওয়ার ব্যবহার করে বানানো, জিপিএল মোতাবেক এই নতুন বদলে তোলা সফটওয়ারটাও আপনা থেকেই জিপিএল-এর আওতায় চলে আসবে। এবার, যে এটা বদলালো সে এটা বিক্রি করতেই পারে। কিন্তু বদলানো সংস্করণ অন্য যে কেউ যদি বিনামূল্যে দেয় তাকে আটকানোর অধিকার তার থাকবে না।

অর্থাৎ, জিপিএল-এর মধ্যে বাজারের চালু কাঠামোর বিরুদ্ধে লড়াই-এর একটা বুদ্ধিদীপ্ত বজ্জাতি আছে — যে কারণে তিনি গোটা ওপন সোর্স জগতের কাছে ধন্যবাদার্হ। একবার জিপিএল-এর আওতায় কোনো সফটওয়ার আসা মানেই সেখান থেকে এটা শেষহীন শৃঙ্খল চালু হয়ে যাওয়া। তাই, একবার জিএনইউ-র কোনো সফটওয়ার ব্যবহার করা মানে গোটা লিনাক্স সফটওয়ারের জগতটাই এসে গেল জিপিএল-এর আওতায়। লিনাস টরভালডস-এর দেওয়া ওই অপারেটিং সিস্টেম সহ। তৈরি হল গ্নু-লিনাক্স। কিন্তু গ্নু-লিনাক্স-এর এই পুরো ঘটনাটাই অনেক পরের। চতুর্থ জেনারেশন কম্পিউটারের সময়ে এসে। পাঁচ নম্বর দিনে এবং পরেও আবার এই প্রসঙ্গে আসব আমরা।

১১।। চতুর্থ জেনারেশন কম্পিউটার — ১৯৮০ থেকে চলছে ...

তৃতীয় জেনারেশন থেকে এবার চতুর্থ জেনারেশন কম্পিউটার আসায় একটা বড় ভূমিকা নিল এলএসআই বা লার্জস্কেল ইন্টিগ্রেটেড সার্কিট। যেখানে এক একটা চিপের এক বর্গ সেন্টিমিটার সিলিকনে ভরা থাকে হাজার হাজার ট্রানজিস্টর। এই এলএসআই-এর হাত ধরেই পৃথিবীতে এল পার্সোনাল কম্পিউটার বা পিসির জমানা। গঠন কাঠামোর দিক থেকে এই পিসি, প্রথমে যাকে ডাকা হত মাইক্রোকম্পিউটার বলে, এর আগের তৃতীয় জেনারেশনের মিনি-কম্পিউটারদের থেকে কিন্তু আদৌ তেমন আলাদা ছিলনা। প্রাথমিক নাটকীয় পার্থক্যটা এল দামের নিরিখে। একটা মিনি কিনতে একটা কোম্পানির একটা বিভাগ নিদেন বিশ্ববিদ্যালয়ের একটা ডিপার্টমেন্ট লাগতই, সেখানে এই মাইক্রোপ্রসেসর চিপের কল্যাণে একজন ব্যক্তিও এবার একটা কম্পিউটার কেনার কথা ভাবতে শুরু করতে পারল।

১৯৭৪-এ, ইনটেল এল তার ৮০৮০ নিয়ে, তার সিপিইউ কাজ করত ৮ বিটে, এবং মোটামুটি ভাবে একটা বিস্তৃত পরিসরের যে কোনো কাজেই ব্যবহার করা যেত ৮০৮০-কে। এই ৮০৮০-র জন্যে ইনটেলের এবার দরকার পড়ল একটা অপারেটিং সিস্টেম বা ওএস। এই ওএস লাগাতে গিয়ে মেশিনে যুক্ত করা হল তখন সদ্য বাজারে আসা ৮ ইঞ্চি ফ্লপি ডিস্ক, তার জন্যে বানানো হল সিপি/এম (CP/M — **C**ontrol-**P**rogram-for-**M**icroarchitecture) নামে একটা ডিস্ক অপারেটিং সিস্টেম। আমরা ডস (DOS — **D**isk-**O**perating-**S**ystem) বলে যাকে চিনি তার তখনো ধরাধামে অবতীর্ণ হওয়ার দেরি আছে। ইনটেল এই গোটা কাজটার দায়িত্ব দিয়েছিল তাদের অন্যতম কনসালট্যান্ট গ্যারি কিল্ডলের উপর। কিল্ডল এবার এই সিপি/এম অপারেটিং সিস্টেমের বিক্রি এবং বিকাশের জন্য নিজের একটা প্রতিষ্ঠান খোলেন, ডিজিটাল রিসার্চ। ১৯৭৭-এর ভিতরেই ডিজিটাল রিসার্চ সিপি/এম-কে সেই সময়ের প্রায় সবগুলো মাইক্রোকম্পিউটারে ব্যবহারযোগ্য করে তোলে। এবার, এই সিপি/এম-এর উপর ভিত্তি করে অনেক অ্যাপ্লিকেশন প্রোগ্রাম বানানো শুরু হয়। শুধু ডিজিটাল রিসার্চ নয়, এতে সামিল হয় আরো অনেক কোম্পানি।

৮০-র দশকের একদম গোড়ায় আইবিএম বাজারে আসে আইবিএম-পিসি নিয়ে। আইবিএম তখন এই পিসির উপযোগী ওএস এবং অ্যাপ্লিকেশন সফটওয়ার খুঁজছে। এর আগেই মাইক্রোসফট-কুখ্যাত বিল গেটস, তার বেসিক (BASIC) নামের ইন্টারপ্রিটার লিখে ফেলেছেন, জন কেমেনি আর থমাস কুর্টজ-এর প্রাথমিক সংস্করণটাকে ব্যবহার করে। মূলত ষাটের দশকে ডার্টমুথ-এর টাইমশেয়ারিং কম্পিউটারগুলোর জন্যে লেখা কম্পিউটার ভাষা বেসিক কাজ করে কম্পাইলার নয়, ইন্টারপ্রিটার দিয়ে। পার্থক্যটা আমরা পরে বুঝব। কম্পিউটার ভাষাকে অকারণ জনপ্রিয় এবং সহজ করতে গিয়ে বেসিক-এ যা করা হয়েছে তা হল, কোনো কোনো প্রোগ্রামিং-শিক্ষকের মতে, পরবর্তীকালে কোনো সিরিয়াস কম্পিউটার ভাষায় কাজ করার ক্ষমতা শেষ করে দেওয়া।

নিজেদের পিসি বাজারে আনার আগে আইবিএম বিল গেটস-এর সঙ্গে যোগাযোগ করে, যদি পিসির উপযোগী কোনো ওএস তার জানা থাকে। যোগাযোগ করা হয় ডিজিটাল রিসার্চের সঙ্গেও। কিন্তু আইবিএম পিসি কিভাবে

পরবর্তী যুগে কম্পিউটার চর্চার নিয়ামক হয়ে উঠবে সেটা অন্য অনেকেরই মত কিল্ডলও আন্দাজ করতে পারেননি। তিনি আইবিএম-কে কোনো পাত্তাই দিলেন না। বিল গেটস দিলেন। অনেকটা সিপি/এম-এর মতই আর একটা ডিস্ক অপারেটিং সিস্টেম বানিয়েছিল সিঅ্যাটল কম্পিউটার প্রোডাক্ট নামে একটা কোম্পানি। যার নাম ডস, ডিস্ক অপারেটিং সিস্টেম। এদের কাছ থেকে গেটস কিনে নিলেন ডস, পরবর্তীকালে মাইক্রোসফটের ব্যবসার আয়তনের তুলনায় আক্ষরিকভাবেই মাত্র খুঁদকুড়োর দামে। মাত্র ৫০,০০০ ডলারে। ওই বেসিক আর এই ডস-কে একসাথে পিসির সফটওয়ার প্যাকেজ হিশেবে আইবিএম-এর কাছে বিক্রি করলেন গেটস। আইবিএম এই মূল প্যাকেজে কিছু বদল ঘটাতে চাইল, তার জন্য গেটসকে লোক ভাড়া করে আনতে হল সেই সিঅ্যাটল কম্পিউটার প্রোডাক্ট থেকেই। এই বদলানো সংস্করণের নাম দেওয়া হল এমএস-ডস। মাইক্রোসফট-ডস। অচিরেই আইবিএম তথা অন্য পিসির দুনিয়া জয় করে নিল এমএস-ডস। মূলত তার প্রায়োগিক সহজতার জন্যে। এখানে কাজ করল গেটসের দ্বিতীয় ব্যবসায়-বুদ্ধি। তিনি এমএস-ডসকে বিক্রি করতে শুরু করলেন বিভিন্ন কম্পিউটার নির্মাতার কাছে, যার কাছে হেরে গেল কিল্ডলের রণনীতি, নির্মাতাকে না-করে, সরাসরি ব্যবহারকারীকে বিক্রি করতে চেয়েছিলেন।

৮১-র পিসি/এক্সটি-র জায়গায় ১৯৮৩-তে আইবিএম বাজারে আনল পিসি/এটি, যাতে মাইক্রোপ্রসেসর ব্যবহৃত হল ইনটেল ৮০২৮৬, এবং ব্যবহৃত হল এটি-বাস, যাতে তথ্য চলাচল করবে ১৬ বিটের এককে, মানে পিসি/এক্সটি-র চেয়ে উন্নততর (AT — **A**dvanced-**T**echnology)। এমএস-ডস ইতিমধ্যেই সুপ্রতিষ্ঠিত এবং সিপি/এম প্রায় অবলুপ্ত। এর পরে ইনটেল যে ৮০৩৮৬ এবং ৮০৪৮৬ মেশিন আনল, তাতেও ব্যাপকভাবে ব্যবহৃত হল এমএস-ডস। এমএস-ডসের প্রথম দিককার সংস্করণগুলো ক্রমে অনেক সংস্কৃতও হল। অনেক উন্নত বৈশিষ্ট্য তুলে আনা হল ইউনিক্স থেকে।

১২।। গ্রাফিকাল ইউজার ইন্টারফেস এবং নেটওয়ার্কিং

এই সিপি/এম বা এমএস-ডস, ঠিক গ্নু-লিনাক্স এর ব্যাশ শেলের মতই, কমান্ড প্রম্পটে কাজ করতে থাকা একটা কমান্ড ইন্টারপ্রিটার। একটা কমান্ড লাইন থাকে, যেখানে একটা কার্সর দপদপ করতে থাকে। সেখানে কিবোর্ড থেকে কমান্ড টাইপ করে তোলা হয়, তারপর এন্টার চাবিটা টেপা মাত্র কমান্ডটা পৌঁছে যায় কম্পিউটারের কাছে — কম্পিউটার কাজ শুরু করে। এই কমান্ড-প্রম্পট থেকে ইঁদুর আর ছবির রাজত্ব মানে গ্রাফিকাল ইউজার ইন্টারফেস বা জিইউআই বা গুই এসেছিল প্রাথমিক ভাবে একটি মানুষের গবেষণার উপর নির্ভর করে। স্ট্যানফোর্ড রিসার্চ ইনস্টিটিউটের ডাগ এঙ্গেলবার্ট ষাটের দশকের মাঝামাঝি তৈরি করেন এই গুই। এর মধ্যে পড়ে উইনডোজ বা কাজ করার জন্যে একটা জানলার ছবি, তার সঙ্গে আরো নানা উপাদান, মেনু, আইকন, বোতাম ইত্যাদি — মাউস ব্যবহার করার নানা তরকিব। এঙ্গেলবার্ট-এর এই কাজকে ঘিরে তারপর ব্যাপক গবেষণা এবং বিকাশের কাজ শুরু হয়। অ্যাপল কোম্পানির স্টিভ জবস এই গুই নিয়ে আসেন অ্যাপল-এ এবং তৈরি হয় অ্যাপল ম্যাকিনটশ। গুই পিসি-র ইতিহাসে আর একটা অধ্যায় নিয়ে আসে, যার অন্য নাম ‘ইউজার ফ্রেন্ডলি’। মানে, ব্যবহারকারী কম্পিউটার বিজ্ঞান কিছু না-জেনেই, ব্যবহারের কিছু রীতি শিখে কম্পিউটার ব্যবহার করতে পারবে, সফটওয়ার প্রয়োগ করতে পারবে তার বিভিন্ন কাজে। এই ব্যবহারকারী শুধু যে কম্পিউটার বিজ্ঞান কিছু জানে-না তাই নয়, জানার কোনো বাসনা বা পরিকল্পনাও তাদের নেই। অর্থাৎ প্রথম যুগের মেশিন যারা ব্যবহার করত সেই প্রোগ্রামার-অপারেটর ইউজার থেকে সম্পূর্ণ নতুন ধরনের একটা ব্যবহারকারী তৈরি হল এবং বাড়তে লাগল — কনজিউমার-ইউজার বা ভোক্তা-ব্যবহারকারী। একটা ছোট সংখ্যক প্রোগ্রামার-ইউজারের দায়িত্ব রইল সফটওয়ার বানানো বদলানো বোঝা এবং চর্চা। আর বৃহত্তর গোষ্ঠী তৈরি হল কনজিউমার-ইউজারের, যারা টিভি-সিডি-প্লেয়ার-ডিভিডি-প্লেয়ার-গেমস-কনসোল-নেট-ব্রাউজার ইত্যাদি বিভিন্ন রকমে ভোগ করবে কম্পিউটারকে, কম্পিউটার বিজ্ঞানের কোনো তোয়াক্কা না-রেখেই।

মাইক্রোসফট যখন এমএস-ডসের পরবর্তী প্রজন্ম নিয়ে কাজ শুরু করল, অ্যাপল ম্যাকিনটোশের জনপ্রিয়তা তাদের ভীষণভাবে প্রভাবিত করল। তৈরি হল একটা গুই-ভিত্তিক ওএস, এমএস-উইনডোজ। প্রথম দিককার উইনডোজ চালাতে হত ডস থেকেই, কমান্ড প্রম্পট-এ ‘উইন’ (win) টাইপ করে এন্টার মারতে হত। উইন্ডোজ তখনো ছিল একধরনের একটা গুই খোলস, ডস ওএস-এর উপরে চলত। প্রায় দশ বছর ধরে, ১৯৮৫ থেকে ১৯৯৫ এই অবস্থা চলেছিল। ১৯৯৫-এ এলো স্বনির্ভর একটা গুই সিস্টেম, উইনডোজ নাইন্টিফাইভ। এখানে আভ্যন্তরীন এমএস-ডস

ব্যবস্থাটা ব্যবহৃত হত বুট করার সময় বা এমএস-ডস কোনো অ্যাপ্লিকেশন চালানোর সময়। বুট করার পরেই পর্দায় ফুটে উঠত উইন্ডোজ-৯৫-এর লোগো, গুই শুরু হয়ে যেত। যদিও, চাইলে, এমএসডস-ডট-সিস (msdos.sys) ফাইলকে বদলিয়ে (Bootgui = 0 থেকে Bootgui = 1) ডস-এর কমান্ড প্রম্পট কনসোলেই শুরু করা যেত।

১৯৯৮-এ মাইক্রোসফট নিয়ে এল উইন্ডোজ নাইন্টিএইট। যা ৯৫-এর ওএস কাঠামোটাকে কিছুটা বদলে দিল। কিন্তু ৯৫ বা ৯৮ দুই উইন্ডোজ-এই ব্যাপকভাবে রয়ে গেল ইনটেল এর ১৬ বিট অ্যাসেম্বলি ল্যাংগুয়েজ প্রোগ্রামিং। এই ৮ বিট ১৬ বিট এগুলো নিয়ে আমাদের আলোচনাটা, শূন্য আর দুই নম্বর দিন থেকে, মনে করুন। ৮ বিট এর চেয়ে ১৬ বিট উন্নততর এবং জটিলতর ব্যবস্থা — কম্পিউটার যখন তার তথ্য পড়া বা লেখা বা মনে রাখার একক হিশেবে ব্যবহার করে ১৬ বিট। যে সময় ১৬ বিট ব্যবস্থা আভ্যন্তরীন ভাবে নিয়ে উইনডোজ ৯৮ বাজারে এল, তখন সময় দাবি করছে উন্নততর ৩২ বিট ব্যবস্থাকে। প্রচুর অ্যাপ্লিকেশন সফটওয়ার এবং ওপন সোর্স জগত তখন প্রায় গোটাটাই ৩২ বিট গঠনে চলে গেছে। এখন, এই ২০০৩ সালে যেমন সময়ের দাবি ৬৪ বিট আর্কিটেকচার। ইতিমধ্যেই একাধিক সার্ভার-মাইক্রোপ্রসেসর এবং পিসি-মাইক্রোপ্রসেসর বাজারে এসে গেছে যারা কাজ করে ৬৪ বিটে। তাদের ওএস এবং অ্যাপ্লিকেশন সফটওয়ারও ইতিমধ্যেই আসতে শুরু করেছে ওপন-সোর্স জগতে।

এর পরবর্তী আর একটা মাইক্রোসফট ওএস হল উইনডোজ এনটি। এনটি এখানে নিউ টেকনলজির সংক্ষেপ। যা এক অর্থে উইনডোজ ৯৫-এর সঙ্গে কম্পেটিবল, অর্থাৎ ৯৫-এর সব প্রোগ্রামই এনটি-তে চালানো যাবে, কিন্তু এনটি হল উইনডোজ জগতে প্রথম পূর্ণাঙ্গ ৩২ বিট ব্যবস্থা। মাইক্রোসফট আশা করেছিল যে এনটি বাজারে এসে পুরোনো সমস্ত এমএস-ডস এবং এমএস-উইন্ডোজ ওএস-কে অপসারিত করে দেবে। কিন্তু সেটা ঘটেনি। মূলত এনটির অত্যধিক পোকাধরা বা বাগড সফটওয়ারের জন্যে। তখন মাইক্রোসফট বাজারে আনে ৯৮-এর একটা নতুন সংস্করণ — উইনডোজ এমই, বা মিলেনিয়াম এডিশন। এটা মাইক্রোসফটের বানিজ্যিক নীতিরই অংশ যে একটা অপারেটিং সিস্টেমের পোকা এবং খুঁত গুলোকে বেছে ঠিক করার জায়গায় আর একটা নতুন নামে আর একটা নতুন অপারেটিং সিস্টেম নিয়ে এসো। এর পোকাগুলো ধরা পড়তে পড়তে আবার নতুন একটা। সেই ভাবে বাজারে এল উইন্ডোজ এক্সপি। উইন্ডোজ এনটিতে কেন্দ্রীয় ঝোঁকটা ছিল সার্ভার মেশিন বা নেট-ওয়ার্ক। আর ৯৫, ৯৮ বা এমই-তে মূল ঝোঁকটা ছিল পিসি। মাইক্রোসফটের কথা অনুযায়ী, উইনডোজ এক্সপি-তে এসে এই দুটো ধারা ফের মিশে গেল।

ওয়ার্কস্টেশন বা উচ্চক্ষমতাসম্পন্ন নেটওয়ার্ক সার্ভার মেশিনগুলোয় ইউনিক্স বা লিনাক্স উইনডোজের চেয়ে অনেক বেশি ব্যবহার হয়। বিশেষত যেসব মেশিনে উচ্চ-শক্তির আরআইএসসি (RISC — **R**educed-**I**nstruction-**S**et-**C**hip) ব্যবহার হয়, সেখানে ইউনিক্স বা গ্নু-লিনাক্সই প্রায় একচ্ছত্র। আরআইএস চিপ হল মাইক্রোপ্রসেসরের এমন এক ডিজাইন যেখানে কম্পিউটেশন প্রক্রিয়াটা অনেক দ্রুত এবং দক্ষ রকমে হয়, একটা খুব ছোট আদেশ-মালা ব্যবহার করে। স্বাভাবিকের চেয়ে ছোট এই আদেশমালাটা তৈরি করা হয় এমনভাবে যাতে এদের ব্যবহার করে অপারেটিং সিস্টেম তথা সফটওয়ার কম্পিউটারের প্রসেসরকে পরবর্তী জটিলতর আদেশগুলো পৌঁছে দিতে পারে। এই ছোট আদেশমালার প্রতিটি আদেশ সেই অর্থে অপটিমাইজড মানে সর্বোচ্চ ক্ষমতায় কাজ করার জন্যে নির্মিত এবং নির্দ্ধারিত, এবং সচরাচর এরা কাজ করে কম্পিউটারের সময়একক এক ক্লক সাইকেলের মধ্যে। আরআইএস চিপরা সরল সোজাসাপ্টা আদেশগুলো সিআইএস (CIS — **C**omplex-**I**nstruction-**S**et) চিপের চেয়ে অনেক বেশি গতিতে করতে পারে। এই সিআইএস চিপ বানানো হয় যাতে তাদের বিভিন্ন রকম কাজে ব্যবহার করা যায় — জেনেরাল পারপাজ। এরা অনেক জটিলতর আদেশও নিতে পারে, কিন্তু তাদের গতি আরআইএস চিপের চেয়ে অনেক কম।

পেন্টিয়াম, অ্যাথলন বা এই গোত্রের অন্যান্য স্বাভাবিক সাধারণ সিপিইউ-র মেশিনের দুনিয়ায়, পিসি-ব্যবহারকারীদের মধ্যে, উইনডোজের সবচেয়ে বড় প্রতিদ্বন্দ্বী এখন গ্নু-লিনাক্স। ২০০৩-এর অগাস্টে, মাইক্রোসফটের এক ঘোষণায় গেটস তাদের সবচেয়ে বড় প্রতিবন্ধক হিশেবে বিশ্ব জুড়ে সামগ্রিক ব্যবসায়িক মন্দার পরই এনেছেন ওপন-সোর্স সফটওয়ারকে। ক্রমশ বিশ্ববিদ্যালয়ের অ্যাকাডেমিক কম্পিউটার ব্যবহারকারীদের গণ্ডি ছাড়িয়ে গ্নু-লিনাক্স ব্যবহার করতে শুরু করেছে বিভিন্ন ব্যবসায়িক প্রতিষ্ঠানও। গ্নু-লিনাক্স ক্রমে অনেক ইউজার ফ্রেন্ডলি-ও হয়ে উঠেছে। বিশেষত গুহনোম কেডিই ইত্যাদি বিভিন্ন রকম গুই-এ কমান্ড প্রম্পট বাদ দিয়ে মাউসে কাজ করার সুযোগ আসায়, এবং মাল্টিমিডিয়া আর গ্রাফিক্সে উইনডোজের তুলনায় যোজন যোজন উৎকৃষ্ট সফটওয়ার তৈরি করায়।

ইউনিক্স এবং গ্নু-লিনাক্সে গুই এসেছিল এক্স-উইনডোজ (X-Windows) নামে, যাকে তৈরি করে এমআইটি, ম্যাসাচুসেটস ইনস্টিটিউট অফ টেকনলজি। যেখানে কমান্ড প্রম্পটের কাজগুলো ছবিতে মাউস টিপে করা যায়। আমাদের এই পাঠমালায় যার খুঁটিনাটিগুলো আমরা আনবনা, আগেই বলেছি। এই পাঠমালায় গোটাটাই আমরা করব কমান্ড প্রম্পটে।

আর উইনডোজের তুলনায় অনেক বেশি সমর্থ বা রোবাস্ট গ্নু-লিনাক্স চিরকালই ছিল। রিস্টার্ট বা হ্যাং-এর দুস্বপ্ন নিয়ে রাতে ঘুমোতে যায়না এমন উইনডোজ ব্যবহারকারী অসম্ভব, আর গ্নু-লিনাক্সে এরা প্রায় অজানা শব্দ। ইন্টারনেট যেহেতু প্রাথমিক ভাবে তৈরি হয়েছিল ইউনিক্স বা গ্নু-লিনাক্সের হাতে এবং জন্যে, স্বাভাবিকভাবেই ইন্টারনেট ব্যবহার করাও উইনডোজের তুলনায় ইউনিক্সে ও লিনাক্সে অনেক সহজ ও সুঠাম। এবং গত কয়েক বছর যেকোনো কম্পিউটার ব্যবহারকারীর কাছে একটা জ্যান্ত ত্রাস হল কম্পিউটার ভাইরাস। কত তথ্য রসদ এবং অর্থ যে বলি হয়েছে এই ভাইরাসে তার কোনো ইয়ত্তা নেই। পরে, গ্নু-লিনাক্স কাঠামোটা আমরা যখন বুঝতে শুরু করব, বিশেষত রুট বা সুপারভাইজার এবং ইউজার বা ব্যবহারকারীর বিভিন্ন অধিকারের পার্থক্যগুলো, তখন বুঝব যে, গ্নু-লিনাক্স এমন করেই তৈরি যাতে ভাইরাসের পক্ষে এই ধরনের কাঠামোয় দাঁত ফোটানো সম্ভব না। আর ইউনিক্স-এর বেলাতেও এগুলো সত্যি, কিন্তু বিভিন্ন কোম্পানির ব্রান্ডনেম প্রোডাক্ট ক্লোজড সোর্স ইউনিক্সগুলো এত দামি যে কোনো সাধারণ ব্যবহারকারীর কাছে, শুধু আমাদের দেশে নয়, পশ্চিমেও, কেনা প্রায় অসম্ভব।

এই চতুর্থ জেনারেশন কম্পিউটারে অপারেটিং সিস্টেমের স্তরে একটা নতুন জিনিষ এসেছিল আশির দশকের মাঝামাঝিতে এসে। যার নাম নেটওয়ার্ক-ওএস বা ডিস্ট্রিবিউটেড-ওএস। এর মূল কারণটা এই যে ক্রমশ আরো বেশি বেশি সংখ্যক ব্যক্তিগত পিসি এবং সার্ভার বা ওয়ার্কস্টেশন পরস্পর সংযুক্ত হয়ে উঠছিল ল্যান ইত্যাদি দ্বারা, মূলত ব্যবসার এবং তথ্য চলাচলের প্রয়োজনে। এই ধরনের ওএস-এ একজন ব্যবহারকারী তার নিজের কম্পিউটার ছাড়া অন্য কম্পিউটারগুলোও ব্যবহার করার সুযোগ পায়, সেখান থেকে কোনো ফাইল কপি করে নিতে পারে নিজের মেশিনে, নিজের মেশিন থেকে ফাইল পাঠাতে পারে অন্য মেশিনে, এবং এই জাতীয় আরো অনেক কিছুর সুযোগ থাকে। অর্থাৎ, গ্লোবাল স্তরে মানে সবগুলো মেশিন মিলিয়ে থাকে একটা নেটওয়ার্ক এবং তার নেটওয়ার্ক ওএস। আর লোকাল স্তরে একটা লোকাল মেশিনে থাকে এক বা একাধিক লোকাল ব্যবহারকারী এবং তাদের একটা লোকাল ওএস। এইখান থেকে সততই পরের পদক্ষেপটা আসে খুব সহজে — ঠিক ওই তৃতীয় জেনারেশনের টাইম-শেয়ারিং-এর মূল ধারণাটা মনে করুন। স্ট্যাটিস্টিক্সের সেই সোনালি নিয়ম অনুযায়ীই, যে কোনো মুহূর্তেই, অনেকগুলো মেশিনের অনেকগুলো প্রসেসর জাস্ট ঝিমোচ্ছে, কারণ তারা তো তাদের ব্যবহারকারীদের মত কফি খেতে বা আড্ডা মারতে পারেনা। তাহলে কেন সেই ঝিমোতে থাকা প্রসেসরদের ঝুলে থাকা সমগ্র শক্তিকে কাজে নিয়ে আসা যাবেনা? যেখানে সব প্রসেসরদের মিলিত ক্ষমতাটা সিস্টেমের প্রসেসিং বা কম্পিউটিং পাওয়ার হয়ে উঠবে? ডিস্ট্রিবিউটেড ওএস যেখানে তার ওই একগুচ্ছ প্রসেসরকে মোট রসদের একটা ভাণ্ডার হিশেবে দেখবে, এবং যখনই যে প্রসেসর তার ক্ষমতার চেয়ে কমে কাজ করছে তাকে নতুন কাজ দেবে। যেভাবে সে অন্য রসদগুলো নিয়ে শেয়ারিং এবং মাল্টিপ্রোগ্রামিং করত। এইভাবেই এল নেটওয়ার্ক ওএস বা ডিস্ট্রিবিউটেড ওএস।

একটা একক লোকাল অপারেটিং সিস্টেমের সঙ্গে ডিস্ট্রিবিউটেড বা নেটওয়ার্ক অপারেটিং সিস্টেমের কিন্তু কোনো মূলগত পার্থক্য নেই। এখানেও একাধিক মেশিনের মধ্যে সংযোগ ঘটতেই পারে। অবশ্যই একটা ল্যান কার্ড বা মোডেম জাতীয় কোনো সংযোগের যন্ত্র থাকবে, এবং সেই যন্ত্র চালানোর জন্যে আর যন্ত্রে আদান-প্রদান করা তথ্যের প্রবাহ নিয়ন্ত্রণের জন্যে থাকবে একটা ড্রাইভার, এবং থাকবে কিছু সফটওয়ার যা দিয়ে দূরবর্তী মেশিনগুলোর সঙ্গে সংযোগ এবং তথ্য দেওয়া-নেওয়া করা যায়। কিন্তু এতে ওএস-এর মৌলিক গঠনের কোনো বদল ঘটেনা। অন্যদিকে, একটা ডিস্ট্রিবিউটেড ওএস-এর একটা মেশিনে বসে ব্যবহার-করার সময় একজন ব্যবহারকারীর কাছে কিন্তু সেটা একটা অভ্যস্ত লোকাল ওএস বলেই মনে হবে। আলাদা আলাদা মেশিনের আলাদা আলাদা হার্ড ড্রাইভ যেন আলাদা আলাদা ডিরেক্টরিতে তথ্যের আলাদা আলাদা ভান্ডার। যদিও এখানে মোট কম্পিউটেশন প্রক্রিয়াটা চলছে অনেকগুলো প্রসেসরকে মিলিয়ে। একজন ব্যক্তি ব্যবহারকারীর জানার বা বোঝার প্রয়োজনও নেই যে আসলে ওএস কোন প্রোগ্রামকে কোন মেশিনের কোন ডিরেক্টরি থেকে তার জন্যে চালাচ্ছে। বা কোন কোন সিস্টেম ফাইল কোন কোন মেশিনের কোথায় রাখা। এই গোটা কাজটাই অভ্যস্ত দক্ষ নিঃশব্দ রকমে করে চলবে ওএস, অবসরহীন দাস।

১৩।। একই সময়ের শরীরে বিভিন্ন সময়

একটা মজার জিনিষ এই যে কম্পিউটারের তথা অপারেটিং সিস্টেমের ইতিহাসে প্রায় একটা উপাদানও কিন্তু আজো অবশেষহীন ভাবে অবলুপ্ত হয়নি। এক একটা নতুন উপাদান পদক্ষেপ আবিষ্কার ঘটেছে এবং তাদের ব্যবহার করা শুরু হয়েছে। আবার এসেছে একটা নতুন কিছু। কিন্তু, তাই বলে পুরোনো উপাদানটা নিশ্চিহ্ন হয়ে যায়নি। এই ২০০৩ সালের মধ্যেই, চতুর্থ জেনারেশন মেশিনের মধ্যেই, প্রায়ই বেশ উজ্জ্বল রেলায়, বেঁচে আছে একটু '৬২, একটু '৭৪, একটু '৮৫ বা একটু '৯৩। ধরুন মেইনফ্রেম বা মিনিকম্পিউটার বা পার্সোনাল কম্পিউটার বা এমবেডেড সিস্টেম — এরা আলাদা আলাদা ভাবে গজিয়ে উঠেছে ইতিহাসের বিভিন্ন মুহূর্তে এবং বিভিন্ন ধরনের জ্যান্ত প্রয়োজনকে মেটাতে। কিন্তু এরা একইসঙ্গে সবাই কাজ করে চলেছে। মোট সভ্যতায়, মোট বাজারে, মোট কম্পিউটার বিজ্ঞানে।

প্রথমদিকের মেইনফ্রেম মেশিনগুলোয় প্রোগ্রাম বানানো হত অ্যাসেম্বলি লেভেল ভাষায়। এমনকি কম্পাইলার বা অপারেটিং সিস্টেম গোছের ব্যাপক জটিল এবং প্যাঁচালো প্রোগ্রামগুলোও লেখা হত অ্যাসেম্বলিতে। মিনি-কম্পিউটার যখন সিনে এল তখন মেইনফ্রেম মেশিনে বহুল ব্যবহার হচ্ছে ফোর্ট্রান বা কোবোল (COBOL — **Co**mmon-**B**usiness-**O**riented-**L**anguage — প্রায় স্বাভাবিক ইংরিজি, ১৯৫৯ থেকে ৬১-র মধ্যে তৈরি হয়, ব্যবসার রোজকার প্রয়োজন মেটাতে) ইত্যাদি হাই-লেভেল ভাষা, মানে মানুষের ভাষার খুবই কাছাকাছি যারা। কিন্তু তখনো, তখন নতুন আসা মিনি-কম্পিউটারগুলোয় প্রোগ্রাম করা হত ওই অ্যাসেম্বলি কোডেই, মূলত মেমরি বা স্মৃতির অভাবের কারণেই। তারপর যখন মাইক্রো-কম্পিউটার মানে প্রথম যুগের পিসি বাজারে এল, তাদেরও প্রোগ্রাম করা হত অ্যাসেম্বলি কোডে। যদিও, মজার কথাটা এই, ততদিনে মিনি-কম্পিউটারে প্রোগ্রাম করা হচ্ছে হাই লেভেল ভাষায়। তার অনেক পরে, নিকট অতীতে, যখন পাম-টপ কম্পিউটার এল, তাদেরও শুরুর দিকে প্রোগ্রাম করা হত অ্যাসেম্বলিতে। স্মার্ট কার্ড ইত্যাদি এমবেডেড সিস্টেম-এর বেলাতেও তাই।

ওএস-এর ক্ষেত্রে, একদম গোড়ার দিকে, মাল্টিপ্রোগ্রামিং-এর জন্য প্রয়োজনীয় প্রোটেক্টিভ হার্ডওয়ার, মানে প্রত্যেকটা রসদ এবং প্রোগ্রামকে তার নিজের নিজের মহলে অন্য প্রোগ্রামের প্রকোপ থেকে বাঁচাতে পারে এমন হার্ডওয়ার, ছিলনা। যেমন আমরা আগেই বলেছি। তাই, তখনকার ওএস একসঙ্গে একটার বেশি প্রোগ্রাম চালাতে পারত-না। মাল্টিপ্রোগ্রামিং তখন সম্ভব ছিলনা। প্রোগ্রামগুলো আবার মেশিনে ঢোকাতে হত হাতে করে। পরে যখন প্রয়োজনীয় হার্ডওয়ার এল, একসঙ্গে একাধিক প্রোগ্রাম চালাতে এবং নিয়ন্ত্রণ করতে শিখল ওএস। তারপর, সময়ের হাত ধরে, নতুনতর হার্ডওয়ারের হাত ধরে, একসময় এল পুরোপুরি পূর্ণমাত্রার টাইমশেয়ারিং। মিনি-কম্পিউটারের প্রথম দিকে, তাদেরও কোনো প্রতিরক্ষামূলক হার্ডওয়ার ছিলনা, তাই সেখানেও মাল্টিপ্রোগ্রামিং করা যেতনা। যদিও, মেইনফ্রেম মেশিনের পৃথিবীতে মাল্টিপ্রোগ্রামিং ততদিনে জলভাত হয়ে গেছে। ক্রমে মিনি-কম্পিউটারেও এল প্রয়োজনীয় হার্ডওয়ার, তার মানে সেই পথে একদিন এল মাল্টিপ্রোগ্রামিং, একই পথে একদিন টাইম-শেয়ারিং-ও এল। পামটপ কম্পিউটার বা স্মার্ট কার্ডের বেলাতেও তাই। অভিযোজনের যে সিঁড়ি বেয়ে একসময় মানুষ নামের প্রজাতিটা এসেছে, প্রতিটি মানুষের ভ্রূণকে তার মায়ের গর্ভে ঠিক যেমন সেই প্রতিটি সিঁড়ি পার হয়ে হয়ে আসতে হয়। তার শরীরে ফুলকার চিহ্ন আসে, আবার মিলিয়ে যায়, সরীসৃপের বুকে হাঁটা অস্তিত্বের পদচিহ্ন আসে, আবার মিলিয়ে যায়, একদিন আবার তারও সন্তান হয়, সে-ও আসে সেই একই পথ পেরিয়ে। বারমুডা আর পুয়ের্তোরিকোর ভিতর সারগাসো সমুদ্রে যেমন ডিম ফুটে বেরোয় ঈল মাছের ছানা, লেপ্টোসেফালি থেকে ঈল হয়, যায় নদীতে, সারাজীবন কাটায় নদীর মিষ্টিজলে, আবার মৃত্যুর আগে ফিরে আসে সারগাসো সমুদ্রের নোনা জলে, ডিম পাড়ে, সেই একই বৃত্ত চলতে থাকে।

হার্ডডিস্ক প্রথম লাগানো হয়েছিল মেইনফ্রেমে, তারপর তা এল মিনিতে, পরে মাইক্রোতে। প্রথম যখন হার্ডডিস্ক এল, তার হাত ধরে এল আদিম ধরনের ফাইলসিস্টেম। ১৯৬০-এর দশক জুড়ে পৃথিবীর সবচেয়ে শক্তিশালী মেইনফ্রেম কম্পিউটার সিডিসি ৬৬০০-তে, একজন ব্যবহারকারী সবচেয়ে বেশি যা করতে পারত তা হল, একটা হার্ডডিস্কে একটা ফাইল তৈরি করা এবং তারপর সেটাকে স্থায়ী বলে ঘোষণা করা, নইলে তার নিজের প্রোগ্রাম থেকে বেরনো মাত্র সেই ফাইল হার্ড ডিস্ক থেকে মুছে যাবে। ওই ফাইল পরে আবার ব্যবহার করার পরিকল্পনা থাকলে প্রোগ্রামের সঙ্গে একটা বিশেষ কমান্ড ব্যবহার করে ফাইলের সঙ্গে একটা পাসওয়ার্ড লাগাতে হত। কারণ, সব যাবতীয় সমস্ত

ব্যবহারকারীর জন্যে ডিরেক্টরি ছিল মাত্র একটা। প্রতিটি ইউজারের প্রতিটি ফাইলই থাকত সেই একমেবাদ্বিতীয়ম ডিরেক্টরিতে। ইউজারদেরই মাথায় রাখতে হত যাতে একজনের দেওয়া একটা ফাইল-নামের সঙ্গে অন্যের দেওয়া অন্য কোনো ফাইলের নাম মিলে না-যায়। পরে যখন আমরা গ্নু-লিনাক্স ফাইলসিস্টেম খুব বিশদে আলোচনা করব সাত আর আট নম্বর দিনে, তখন দেখবেন ফাইলব্যবস্থার সেই প্রারম্ভিক কাঠামোগুলো কী রকম ফিরে ফিরে এসেছে।

ভারচুয়াল মেমরি বলতে আমরা বুঝি মেমরির ভৌত পরিমাণের চেয়ে বেশি স্মৃতিকে ব্যবহার করতে পারা। আমরা উল্লেখ করেছি আগেই, সোয়াপ ফাইলের কথা বলেছি। সোয়াপ-ফাইল ঠিক এইটাই করে, ফিজিকাল বা ভৌত স্মৃতিকে নিজের বাইরে বাড়িয়ে তোলে ভারচুয়াল বা ভৌতিক স্মৃতি বানিয়ে। ধরুন আপনার মেশিনে ২৫৬ এমবি র‍্যাম আছে, মানে স্মৃতির ভৌত পরিমাণটা হল ২৫৬ এমবি মানে ২৫৬ × ১০২৪ × ১০২৪ × ৮ = ২,১৪৭,৪৮৩,৬৪৮ সংখ্যক বিট বা খোপ — যার প্রতিটি খোপে আসতে পারে হয় একটা ১ অথবা একটা ০। এবার, সোয়াপ ফাইল কাকে বলে মনে করুন, হার্ড ডিস্কে অস্থায়ী ভাবে কিছু লিখে রাখার একটা জায়গা। এবার ধরুন আপনার মেশিনে সোয়াপ ফাইলের সাইজ ৫১২ এমবি। কোনো প্রোগ্রাম চালানোর সময় প্রোগ্রামের সাইজ, প্রোগ্রামের জন্য ব্যবহৃত মোট তথ্যের সাইজ সব মিলিয়ে সীমাটা নির্দিষ্ট ছিল ২৫৬ এমবি। এখন সোয়াপ ফাইলের দৌলতে অন্তত তার দ্বিগুণ আয়তনের মোট প্রোগ্রাম ও তথ্য আপনি ব্যবহার করতে পারবেন। অর্থাৎ আপনার মেশিনের ফিজিকাল মেমরি ২৫৬ এমবি-কে আপনি অতিক্রম করে যেতে পারলেন ভারচুয়াল মেমরি বা ভৌতিক স্মৃতির দৌলতে। হার্ড ডিস্কে অস্থায়ী ভাবে লেখা ওই সোয়াপ ফাইল তখন আর এক রকম র‍্যামের কাজ করছে। ঠিক কী ভাবে এই বাড়িয়ে তোলাটা ঘটে, সেটা আমরা পরে বুঝব।

এই প্রসঙ্গে একটু বলে নেওয়া ভাল — পরে আমরা আরো বিশদে বুঝব, আমাদের মেশিনে স্মৃতি বা মেমরি আসলে পাঁচ রকমের। এবং প্রোগ্রাম থেকে প্রোগ্রামে মেশিনের গতির যে বিস্তর ফারাক ঘটে তার একটা বড় কারণ এই পাঁচ ধরনের স্মৃতির নিজেদের ভিতরকার গতি এবং আকারের তফাত। এই পাঁচ ধরনের স্মৃতি হল :

(১) মাইক্রোপ্রসেসরের নিজের ভিতরকার রেজিস্টার

(২) আভ্যন্তরীন বা ইন্টারনাল ক্যাশে, মানে, চিপের শরীরেই থাকা স্মৃতি-প্যাকেট

(৩) চিপের বাইরে স্মৃতি-প্যাকেট বা এক্সটারনাল ক্যাশে

(৪) মূল মেমরি মানে যাকে সচরাচর আমরা র‍্যাম বলে ডেকে থাকি

(৫) হার্ড ডিস্ক

ইন্টার্নাল এবং এক্সটারনাল এই দুই ধরনের ক্যাশে (cache) বা স্মৃতি-প্যাকেট হল ছোট ছোট মেমরির প্যাকেটের মত যেখানে কোনো তথ্য বা প্রোগ্রামের পুরোটা বা অংশ মেশিন রাখে যাতে তাদের পাওয়ার সময় হার্ড-ডিস্ক থেকে পড়ার সময়টুকুও ব্যয় করতে না-হয়। ক্যাশে মেশিনের কাজের গতিকে বাড়িয়ে তোলে, কিন্তু একই সঙ্গে, এখনই যা দেখব আমরা, বাড়িয়ে তোলে খরচের পরিমাণও। এবার ইন্টারনাল ক্যাশে হল তাই যার আবাস চিপের ভিতরেই, আর এক্সটার্নাল ক্যাশে মানে তার আবাস চিপের বাইরে। এখানে একটা কথা বলে রাখা ভাল — মেমরির তুলনায় সিপিইউ অনেক বেশি দ্রুত বলেই কিন্তু ক্যাশে ব্যবহার করতে হয়। গল্পটা অনেকটা ভুতের হাতের লেজের মত, নিদ্রা চটিয়ে তাকে তোলার পর, কাজ দেওয়াটাই হয়ে দাঁড়িয়েছিল সমস্যা, শেষে একটা কুকুরের বাঁকানো লেজ তাকে সোজা করতে দেওয়া হয়েছিল। এখানে আবার সেটাও করার জো নেই, খুবই কস্টলি ভুত। তাই, সেই দ্রুতগতিতে কাজ করে চলা রাজমিস্ত্রির সঙ্গে সাধারণের চেয়ে একটু দ্রুততর যোগাড়ে লাগানো হয়, যার নাম ক্যাশে। এরা এমনি যোগাড়েদের পাশাপাশি কাজ করে এবং যোগানোর কাজটাকেই মোটের উপর দ্রুততর করে তোলে। তুলনামূলক ভাবে শ্লথ মেমরিকে দ্রুততর করে তোলা হয়, সিপিইউ-র কাজ করে চলার স্বার্থে তথ্যরাজি স্থানান্তরের যে মোট কাজ থাকে মেশিনে, সেই কাজটাই করে ক্যাশে, আর একটু বেশি গতিতে। মেমরির সঙ্গে প্যালা দেয়। টেকনোলজি তো শুধুই বদলাচ্ছে। কোনোদিন যদি দেখা যায়, মেমরি সিপিইউ-র চেয়ে দ্রুততর হয়ে গেছে, তখন আর ক্যাশে ব্যবহারের কোনো প্রয়োজন পড়বে না।

(১) থেকে (৫) — উপরের এই তালিকাটা করা হয়েছে সবচেয়ে দ্রুতগতি স্মৃতি থেকে সবচেয়ে শ্লথগতি স্মৃতির দিকে। চিপের নিজস্ব রেজিস্টার সবচেয়ে তাড়াতাড়ি কাজ করে — সেকেন্ডে মোটামুটি দশহাজার কোটি বার মেশিন একে পড়তে পারে, এক নম্বর দিনে আমরা এই রেজিস্টারগুলো নিয়ে কথা বলেছি, মনে পড়ছে? আর সবচেয়ে শ্লথগতি হল হার্ডডিস্ক — সেকেন্ডে মোটামুটি একশোবার মেশিন একে পড়তে পারে। তাও, এই হিশেবটা পুরোনো, খুব আধুনিক মেশিনে গতিটা এর চেয়েও অনেকটাই বেশি। কিন্তু গতি বাড়ানো মানেই খরচ বাড়িয়ে তোলা। যে কোনো একটা সময়ে কোনো একধরনের কম্পিউটারে কোন ধরনের স্মৃতি কতটা থাকবে সেটা ঠিক হয় সেই মুহূর্তের বিভিন্ন প্রকৌশলের দাম আর ওই মেশিনে যে কাজ করা হবে তার প্রকৃতি ও পরিমাণের উপর দাঁড়িয়ে। আমি আমার মেশিনকে ব্যবহার করি মূলত লেখার কাজে, আর একটু একটু প্রোগ্রাম শিখতে গিয়ে লেখার কাজে, সেটাকেও প্রোগ্রামিং না বলে লেখা বলাই ভালো। আর গান শোনার বা ফিল্ম দেখার কাজেও ব্যবহার করি, কখনো ছবি আঁকতেও, এই বইয়ের ছবিগুলো যেমন। তাই, আমার যে মেশিনে চলে যায়, থ্রিডি অ্যানিমেশনের কাজ করে এমন কারুর তো তাতে চলবেনা।

যাইহোক, যে কথা আমরা বলছিলাম, অপারেটিং সিস্টেমের বিবর্তনের বিষয়ে — সেটা এই ভারচুয়াল মেমরির ক্ষেত্রেও সত্যি। স্মৃতির ভৌত পরিমাণের চেয়ে বড় আকারের স্মৃতির প্রয়োজন মেটাতে ভারচুয়াল বা ভৌতিক স্মৃতি প্রথম এসেছিল মেইনফ্রেম কম্পিউটারে। তারপর এল মিনিতে। তারপর মাইক্রোয়। ক্রমে ছোট থেকে ছোটতর মেশিনে। নেটওয়ার্কিং-এর ইতিহাসও এই একই।

তিন নম্বর দিনের ভূমিকায় যা আমরা বলেছিলাম, সফটওয়ার কী পথে বিকশিত হবে, সেটা নির্ভর করে সেই যুগের মানবসমাজের কাজের প্রয়োজনের উপর। আর সফটওয়ারের বাস্তব বিকাশ সবসময়ই নির্ভর করে থাকে সেই যুগের হার্ডওয়ারের উপর। প্রথম দিকের মাইক্রো-য় মেমরি ছিল মাত্র ৪ কেবি, আর কোনো প্রতিরক্ষামূলক হার্ডওয়ার তাতে ছিলনা। ওই রকম কোনো মেশিনে হাইলেভেল ভাষা বা মাল্টিপ্রোগ্রামিং ব্যবহার করা অসম্ভব। তারপর মাইক্রোরা বদলাতে বদলাতে আজকের পিসি হয়ে উঠল, তাদের হার্ডওয়ার বদলে গেল, তখন বদলাল তাদের সফটওয়ার সমাহারটাও। আজ তিরিশ বছর আগের মেইনফ্রেমের গল্প শুনে অবাক হই, কিন্তু, এই বদল আরো বাড়বে। আরো তিরিশ বছর বাদে অন্য কেউ যদি পড়ত এই লেখাটা, পড়বেনা, কারণ বিংশ শতাব্দীতে মানুষের শোকের আয়ু একবছর, ওয়েবপেজ বড়জোর দুতিনশো মিনিট, তার কাছে আমাদের এই অবাক হওয়াটা খুব অবাক লাগত। হাসত সে। মনে হত, ওই পৌরাণিক সিস্টেম নিয়ে এরা এত উচ্ছ্বাস করেছে? যেমন আমরা সিঁদুরচরণের কথা পড়ে হাসি, বনগাঁর গ্রাম থেকে নদীয়া অব্দি গিয়ে হতভম্ব হয়ে যে ভেবেছিল, 'এই পিথিমির কি কোনো সিমেমুড়ো নেই?'

ইউনিক্স-এর দুটো বই, দুটোরই চ্যাপ্টার এক, আমাদের খুব কাজে আসবে। একটা সুমিতাভ দাস-এর, ইউনিক্স - দি আল্টিমেট গাইড। আর অন্যটা কারনিঘান অ্যান্ড পাইক-এর ইউনিক্স - দি প্রোগ্রামিং এনভায়রনমেন্ট। আর, ট্যানেনবম-এর মডার্ন অপারেটিং সিস্টেম তো আছেই। যদি কেউ এই স্মৃতির ব্যাপারটা বেশ বিশদে পড়তে চায় তার জন্যে জিএলটির সিডির 'টোটাল হাউ-টু ফ্রম টিএলডিপি' থেকে 'ইউনিক্স অ্যান্ড ইন্টারনেট হাউ-টু'-টা বেশ ভাল টেক্সট। এর পরের দিন আমরা সরাসরি ঢুকব গ্নু-লিনাক্স একটা সিস্টেমে, তার বুট প্রক্রিয়া থেকে শুরু করে।

glt-mad@ilug-cal.org

MADHYAMGRAM

গ্নু-লিনাক্স ইশকুল

জিএলটি মধ্যমগ্রাম
glt-mad@ilug-cal.org

কম্পিউটার, তার হার্ডওয়ার সফটওয়ার, ইতিহাস, অপারেটিং সিস্টেম এবং অ্যাপ্লিকেশনের প্রাথমিক আলোচনা সেরে, অবশেষে আমরা গ্নু-লিনাক্সে পৌঁছচ্ছি এবার, যেখানে পৌঁছনোর কাজেই আমাদের এই জিএলটি ইশকুল আর তার ক্লাস। তবে ভাবার কোনো কারণই নেই যে এখান থেকে আমরা গ্নু-লিনাক্সাধিকারী হয়ে গেলাম। যেটা ম্যাক্সিমাম ঘটতে পারে এই ক্লাস থেকে তা হল, কীভাবে এগোবো তার একটা আন্দাজ — এগোতে হবে নিজে, এগোতে গিয়ে কেলো ঘটবে, সিস্টেম ঘেঁটে যাবে, ইনস্টলেশন নষ্ট হয়ে যাবে — যত প্রভূত পরিমাণে হবে এবং হতে থাকবে এইসব যাবতীয় কেলেঙ্কারি, ততই আসলে শেখাটা এগোবে। আর উইন্ডোজ তো হল জর্জ শার্দুলের মন্তব্যে হলিউড ছবির মত — চুইংগাম চিবোতে চিবোতে অ্যাভারেজ অ্যামেরিকানও যাতে তরতরিয়ে এগোতে পারে। অ্যাভারেজ অ্যামেরিকান বোধহয় পৃথিবীর সবচেয়ে ভয়ঙ্কর অ্যাভারেজ, মূর্খতা অ্যাফর্ড করার জন্যেও তো সম্পদশালী হতে হয়। সেই উইন্ডোজ থেকে যারা গ্নু-লিনাক্সে অভিবাসী হয়, তাদের আরো একটা সমস্যা, ডায়োরিয়ার রিফ্লেক্সকে আদ্যোপান্ত বদলে ফেলতে হয় কনস্টিপেশনের রিফ্লেক্সে। অবশ্য এর একটা চুইংগাম সমাধানও আছে — গুইতে কাজ করা, মাউসের পেট টেপার বাইরে আর প্রায় কোনো ক্রিয়ায় না-গিয়ে। কিন্তু এর মানে, আপনার নাগালের মধ্যে ছিল একটা দৈত্য — যেকোনো কাজ তাকে দিয়ে করাতে পারতেন, না-করে, শুধু তার শিঙে একটু পিঠ চুলকে নিলেন। মাইক্রোসফট আপনাকে এই সুযোগ দিতে পারেনা, ওটা তাদের ব্যবসা — কিন্তু গোটা গ্নু-লিনাক্স কমিউনিটিটা, চিন থেকে মেক্সিকো, আইসল্যান্ড থেকে অস্ট্রেলিয়া, আপনার জন্যে বসে আছে, তাদের করে-ফেলা কাজ এবং আরো কাজের সম্ভাবনা নিয়ে। আপনাকে নেমন্তন্ন করছে, বাইটের পর বাইটে — শেখো, প্রয়োগ করো, সব বই খাতা নোটবই তোমার সামনে খোলা। আপনি শিখতে যান, ঘেঁটে ফেলুন, যদি নেটে ঢোকার পয়সা থাকে তো সোজা লাগ-এর, মানে লিনাক্স-ইউজার-গ্রুপের সাইটে চলে যান, www.ilug-cal.org। আর যদি না-থাকে, ওয়েল অ্যান্ড গুড, জিএলটিতে আসুন, আমরা যেটুকু জানি তাই দিয়ে ঘাঁটা মেশিনকে সিধে করার চেষ্টা চালান, তাতেও না-হলে লাগ-এ, তারপরেও না-হলে নেটে। গোটা গ্নু-লিনাক্স গ্লোব আপনার জন্যে দিনে তেইশ ঘন্টা উনষাট মিনিট ষাট সেকেন্ড জেগে রয়েছে অতন্দ্র। তারপর, যখনই একটু শিখলেন, মনে রাখবেন, আপনার কমিউনিটিতে, ঠিক যেমন আপনি পেলেন, আপনারও দেওয়ার আছে। পশ্চিমের লিনাক্সীদের চেয়েও, বোধহয় আপনার দায়িত্ব একটু বেশি, আপনার চার পাশে এত এত বেশিসংখ্যক তৃতীয় বিশ্বের তৃতীয় ছানারা, যাদের কোনোকিছুরই অধিকার নেই, খাওয়ার অধিকার, পরার অধিকার, কম্পিউটার শেখার অধিকার। সেই গ্নু-লিনাক্স আজ এল, শেষ অব্দি, জিএলটি পাঠমালার এই সিরিজে। আজ প্রথমে বুটপ্রক্রিয়াটা আলতো করে একটু, তারপর লগ-ইন। কম্পিউটার বুট করার পর গ্নু-লিনাক্স সিস্টেমে কাজ শুরু করার আগে নিজের পরিচয় দিয়ে নিতে হয়, যার ভিত্তিতে সিস্টেম ঠিক করে এই ব্যবহারকারীর অধিকার কতদূর অব্দি। কোন ফাইল সে পড়তে লিখতে কাজে লাগাতে পারবে, কোন ডিরেক্টরিতে সে ঢুকতে পারে — ইত্যাদি। গুই মোডেও ঢোকা বা বেরোনো যায় একটা সিস্টেম থেকে। কিন্তু পিছনে কাজটা হয় কমান্ড দিয়েই — শুধু দেখানোর জন্যে থাকে ইঁদুরের কুটকুট, হাতির যেমন দুসেট দাঁত, দেখানোর আর খাওয়ার। তাই, কমান্ড প্রম্পটে লগ-ইন-এর ধারণাটাই আসল। তারপর যাব শেলে, ব্যাশ শেল কী করে অন্য সব প্রোগ্রাম আর কারনেলের মধ্যে মধ্যস্থতা করে।

।। দিন পাঁচ।।

১।। ইউনিক্স — মেশিননিরপেক্ষ অপারেটিং সিস্টেম

আর একবার একটু ফেরত যাওয়া যাক ইতিহাসে। সেই দানবিক মেশিনগুলোকে মনে করুন। হলঘরের বাড়ির স্টেডিয়ামের মত সাইজ। শুধু এই আখাম্বা সাইজটাই না, সবচেয়ে বড় সমস্যাটা এই যে, প্রতিটি মেশিনেই কাজ করছে আলাদা আলাদা সিস্টেম। প্রত্যেকটা মেশিনের নিজস্ব এবং স্বতন্ত্র। একটা মেশিনে চলা প্রোগ্রাম অন্য মেশিনে চলবে না। প্রতিটি প্রোগ্রামকে মেশিনমাফিক বদলে, কাস্টমাইজ করে নিতে হত অন্য মেশিনে চালাতে গেলে। প্রতিটি প্রোগ্রাম তথা সফটওয়ারের বেলাতেই এই একই ঝঞ্ঝাট। সাধারণ ইউজারের, সিস্টেম অ্যাডমিনিস্ট্রেটরের, প্রোগ্রামচির, সবার ঝামেলা। আর, একটা কম্পিউটার রাখা মানে তখন হাতিপোষা। কেনার খরচ, চালানোর খরচ,

রক্ষণাবেক্ষণের খরচ, প্রতিবার প্রতিটি সফটওয়ারকে করে বদলে ফেলার খরচ, সাধারণ ইউজারকে ওই প্রোগ্রাম শিখিয়ে নেওয়ার খরচ। এই ভাবে চলেছিল অনেকগুলো দিন। ১৯৬৯ নাগাদ, বেল ল্যাবরেটরির কেন থম্পসনদের টিম হাত দিল এই ঝামেলা মেটানোর কাজে। মূল ইসুটা খেয়াল করুন — কম্পেটিবিলিটি বা সাযুয্য। একটা মেশিনের জিনিষপত্তর যাতে মেশিন বদলানোর সময় ফেলে দিতে না-হয়। প্রতিটি মেশিনে প্রতিবার প্রতিটি সফটওয়ারকে কেচে গণ্ডুষ করার এই রেকারিং কেলো থেকে মুক্তি। যে কোনো মেশিনের জিনিষপত্তর যাতে অন্য মেশিনে চালানো যায়।

থম্পসনরা তাদের কাজটা শুরু করলেন একদম গোড়া মানে অপারেটিং সিস্টেমের স্তর থেকে। তারা একটা নতুন সিস্টেম বানালেন, যার নাম ইউনিক্স। আগের অপারেটিং সিস্টেম থেকে এটা তিনটে মূল জায়গায় আলাদা।

> (১) চালানো এবং বোঝাটা বেশ সরল এবং সুদৃশ্য, সিম্পল এবং এলিগান্ট। সুদৃশ্য বলতে এখানে স্পষ্ট বুদ্ধিদীপ্ত, সহজে বোধযোগ্য, যৌক্তিক পদক্ষেপে পরপর বিন্যস্ত। এই অর্থে এলিগান্ট শব্দটা প্রায়ই ব্যবহার হয় কম্পিউটার শাস্ত্রে। প্রোগ্রামের বেলাতেও। সিম্পল এবং এলিগান্ট হওয়ার মূল সুবিধেটা এই যে, কিছুটা মনে রাখতে পারলে, তারপর ভেবে ভেবেই কাজ করা যায়, গোটা নিয়মকানুন মুখস্থ করতে হয়না। কাজের একটা এলাকায় যে ধরনের পদক্ষেপে যে ধরনের যুক্তিমালা ব্যবহার হল, সেটা অন্য জায়গাতেও হয়, তাই একবার বুঝে গেলেই সেটা নানা জায়গায় কাজে লাগানো যায়।
>
> (২) আগেকার অ্যাসেম্বলি কোড থেকে সরে এসে সরাসরি সি নামের কম্পিউটার ভাষায় লেখা হল এই অপারেটিং সিস্টেম। এতে মূল মজাটা এই যে, আগের ওই অ্যাসেম্বলি কোড সাধারণভাবে অপাঠযোগ্য। জানিনা সত্যি কিনা, আমি এই গল্প রিচার্ড স্টলম্যান এবং অ্যালান কক্স দুজনের সম্পর্কেই শুনেছি, এরা নাকি অ্যাসেম্বলি কোড পড়ে প্রোগ্রাম বুঝতে পারে। হতেই পারে এটা একটা শাহরিক রূপকথা, আর্বান লেজেন্ড। যদি নাও হয়, তাহলেও সেটা কোনো কাজের কথা না। অসম্ভব পরিশ্রম এবং সময়ের অপব্যয়। একটা গোটা ফিল্মকে স্থির স্লাইডে দেখার মত, সেকেন্ডে চোদ্দটা করে ফ্রেমের স্লাইড ধরে ধরে চাইলে একটা ফিল্মকে তো দেখাই যায়। এতদিনকার চলে আসা সেই অ্যাসেম্বলি কোডের বদলে এবার প্রোগ্রামগুলোকে সি দিয়ে স্টেপ-বাই-স্টেপ লেখা পড়া এবং বোঝা যাবে। কোথাও কোনো গড়বড় হলে, সংশোধন করা যাবে। মানববোধ্য হয়ে উঠল গোটাটা।
>
> (৩) বোধযোগ্য মানবিক হাইলেভেল কোডে লেখা মানেই আর একটা বড় নাটকীয় বদল কিন্তু ঘটে গেল। এবার থেকে ওই অপারেটিং সিস্টেম রিসাইকল করা যাবে, একটা মেশিন থেকে অন্য মেশিনে নিয়ে যাওয়া যাবে, সেই মেশিনে নতুন করে কম্পাইল করে নেওয়া যাবে অপারেটিং সিস্টেমের কোডটা। সেই কম্পাইল করতে গিয়ে যদি দেখা যায়, কোনো সমস্যা ঘটছে, প্রয়োজনমাফিক তাকে বদলানো যাবে, সহজেই। কারণ, সেটা তো সি নামক মানববোধ্য ভাষায় লেখা। একটা মেশিনের উপাদান বা ডিভাইস অন্য মেশিনের থেকে আলাদা। তার মানে আলাদা ডিভাইস ড্রাইভার, আলাদা আলাদা রকমের সফটওয়ার আদেশ। উপাদান মোতাবেক আলাদা আলাদা ড্রাইভার কোডের মধ্যে ঢুকিয়ে বা বাদ দিয়ে নতুন মেশিনে কম্পাইল করে নিলেই অপারেটিং সিস্টেমটা নতুন মেশিনে চালানো যাবে। কারণ সেটা সি-তে লেখা। মানে, এই প্রথম আমরা পেলাম পুনর্ব্যবহারযোগ্য অপারেটিং সিস্টেম। মেশিন থেকে মেশিনে যে কোনো প্রোগ্রামের সাযুয্য বা কম্পেটিবিলিটির ভিত্তিটা তৈরি হল। এই পুনর্ব্যবহারযোগ্য অপারেটিং সিস্টেম দিয়ে সাযুয্যের প্রথম শর্তটা পালিত হল। অপারেটিং সিস্টেমটা কম্পেটিবল হয়ে উঠল।

এখানে সবচেয়ে গুরুত্বপূর্ণ এই তিন নম্বর পয়েন্টটা, মানে, রিসাইক্লিং। এর আগে অব্দি বাজারে যত কম্পিউটার বিক্রি হত, তার প্রত্যেকটারই আলপিন-টু-এলিফ্যান্ট গোটাটাই একেবারে নিজের। এই কম্পিউটারগুলোর গোটা অপারেটিং সিস্টেমের গোটা কোডটাই তৈরি হত গোটাটাই একদম ওই মেশিনের জন্যে আলাদা করে বানিয়ে তোলা। একটা মেশিন বিক্রি করা মানেই তার গোটা অপারেটিং সিস্টেমটা লিখে নেওয়া, তার কাজের সঙ্গে মিলিয়ে, যে যে হার্ডওয়ার উপাদান ব্যবহার করা হবে মেশিনে তাদের ড্রাইভারগুলোকে মিলিয়ে, তাদের উপরে যে যে কাজ করা হবে সেই কাজের প্রয়োজনের সঙ্গে মিলিয়ে। সেই গোটাটাই বদলে গেল ইউনিক্সে এসে। ইউনিক্স অপারেটিং সিস্টেমে গোটাটাই হয়ে উঠল মেশিন-নিরপেক্ষ। মানে, যা সব মেশিনেই এক। শুধু একটা ছোট জায়গাকে বাদ দিয়ে। কোন কোন ভৌত উপাদান মেশিনটায় থাকছে, তার ভিত্তিতে কিছু আলাদা মডিউল বা খণ্ডাংশ গুঁজে দেওয়া

হবে অপারেটিং সিস্টেমে বা কারনেলে। ধরুন যে মেশিনে হার্ডডিস্ক নেই, তার হার্ডডিস্ক ড্রাইভারের মডিউল লাগবে না, আবার খুব আলাদা ধরনের কোনো মেমরি থাকলে, তার বিশেষ রকমের মডিউলটা দিয়ে দিতে হবে কারনেলে। দু-একটা কথা বলে নেওয়া যাক অপারেটিং সিস্টেম আর কারনেল নিয়ে।

এই পাঠমালা চলতে চলতেই গোটা বিষয়টা আমাদের কাছে অনেক স্পষ্ট হয়ে উঠবে। একটা বাস্তব কেজো জিনিষকে সংজ্ঞা দিয়ে বোঝা খুব কঠিন। তবু একটা সংজ্ঞা দেওয়ার চেষ্টা করা যাক। কারনেল কাকে বলে? কারনেল হল অপারেটিং সিস্টেমের হৃদপিণ্ড, যার উপর মেশিনের ভৌত যন্ত্রাংশগুলো সহ যাবতীয় উপাদান নির্ভর করে থাকে। এটা তো প্যালার সেই 'লাথি মানে জাড্যাপহ' — সেইরকম হল। ব্যাপারটা কী? কারনেল কী করে? কারনেল দেখভাল করে একদম নিম্নতম স্তরে ভৌত যন্ত্রাংশগুলোর একটার সঙ্গে অন্যটার সম্পর্ক, নিয়ন্ত্রণ করে মেশিনের রসদ-বন্টণ, যার মধ্যে পড়ে মেমরি তথা হার্ডডিস্ক সহ যাবতীয় স্মৃতিভূমি, এবং তথ্য ঢোকা-বেরোনো বা ইনপুট-আউটপুট। মানে, কোন ব্যবহারকারী ডিস্কভূমির কোন কোন অংশে নিজের পদচিহ্ন রাখতে পারবে, সেটাও শেষ অব্দি নিয়ন্ত্রণ করে কারনেল। কী ভাবে সেটা বুঝতে আমাদের আট নম্বর দিনের ফাইলসিস্টেম অব্দি সবুর করতে হবে। অর্থাৎ সিস্টেমের নিরাপত্তার দেখভালের দায়িত্বও এক ভাবে দেখলে কারনেলের হাতে চলে আসছে। আর মেমরি এবং ইনপুট-আউটপুটের নিয়ন্ত্রণ এবং বন্টন নিয়ে কিছু কথা আমরা ইতিমধ্যেই বলেছি, পরে আরো অনেকটা বলব। কিন্তু একটা অস্বস্তি লাগল না? এইগুলো তো অপারেটিং সিস্টেম বা ওএস-এর কাজ, বারবার বলেছি আগে। তাহলে? কারনেলই কি ওএস?

একটা সিস্টেম, যার ভিতরে কাজ করা যায়, অপারেট করা যায়, যে নিজে অপারেট করে আপনাকে অপারেট করতে দেয়, সেটাই ওএস। তাহলে, মনে মনে ভাবার চেষ্টা করুন। ওএস-এর মধ্যে কী কী থাকবে? হার্ডওয়ার বা ভৌত উপাদানগুলোর সঙ্গে কথা বলতে পারা এবং তাদের ভাগবাঁটোয়ারা করতে পারা। এটা প্রাথমিক। এবার এটার উপর দাঁড়িয়ে প্রয়োগ বা অ্যাপ্লিকেশনগুলো চালানোর একটা জমি বানিয়ে দেওয়া। এই জমি বানানোর জন্যে তার দরকার পড়ে সিস্টেমের কনফিগারেশন তথা কিছু প্রাথমিক প্রোগ্রাম চালানোর ব্যবস্থা করা — যে প্রক্রিয়া বা প্রসেসগুলো না চললে অন্য প্রোগ্রামগুলো চালাতে পারবেনা ইউজার। ইউজারের চালানো ইউজার অ্যাপ্লিকেশনগুলো চলার জন্যেই এই সিস্টেমের অভ্যন্তরে এই প্রক্রিয়াগুলো চলা দরকার।

ধরুন, খুব সহজ, লোকেট বা 'locate' বলে একটা ইউজার কমান্ড দিয়ে কোনো ফাইলকে খোঁজা যায়। কিন্তু এই খোঁজাটা সম্ভব হওয়ার জন্যে প্রথমে গোটা সিস্টেমের যাবতীয় ফাইলের তালিকার একটা ডেটাবেস বানিয়ে নেওয়ার এবং মাঝেমাঝেই সেই ডেটাবেস পুনর্নবীকরণের দরকার পড়ে, কারণ ফাইল তো নিয়তই বদলাচ্ছে সিস্টেমে, কাজ করা মানেই ফাইল বানানো এবং বদলানো। এর কমান্ডটা হল আপডেটডিবি বা 'updatedb'। এবং সিস্টেম যাতে নিজে নিজেই এই ফাইলতালিকা পুনর্নবীকরণের কাজটা সময় থেকে সময়ে করে যেতে পারে তার জন্যে এই কাজটাকে তুলে দেওয়া যেতে সময় মাফিক কাজ করার যখ বা ডিমনের হাতে। এরকম একটা ডিমন বা যখ হল অ্যাট বা 'at'। সত্যিই, অ্যাট সেভেন-থার্টি কিছু করে দিতে বলে রাখলে তাকে তার কখনো ভুল হবেনা, সাড়ে সাতটা বাজা মাত্রই সে তার কাজ শুরু করে দেবে। এবার, মজার কথা হল, এই লোকেট কমান্ডটা যে কোনো ইউজার চালাতে পারে। কিন্তু ওই আপডেটডিবি বা অ্যাট নিয়ে কিছু করার জো নেই তার, শুধু সিস্টেমপ্রভু মানে রুট সেটা পারবে, কারণ, ওইখানে কিছু করতে গেলে কারনেলের স্তরে নিয়ন্ত্রণ লাগবে। এই অ্যাট বা আপডেটডিবি হল সিস্টেমের আভ্যন্তরীন প্রক্রিয়াগুলোর উদাহরণ। নয় এবং দশ নম্বর দিনে গিয়ে আমরা এরকম আরো অনেক সিস্টেম প্রসেস বা সিস্টেম প্রোগ্রামের সঙ্গে পরিচিত হব। তার মানে দেখুন, অপারেটিং সিস্টেমের মধ্যেই অন্তত দুটো স্তরকে খুব সহজে চিনে নেওয়া যাচ্ছে। একটা স্তর ভৌত উপাদানের দুনিয়া। আর একটা স্তর সিস্টেম প্রোগ্রামের। এই প্রথম স্তরটা হল কারনেলের যা কখনো কখনো ঢুকে পড়ে দ্বিতীয়টার মধ্যেও। আর এদের এবং এদের সঙ্গে আরো অনেককিছু মিলিয়ে তৈরি হয় ওএস। যার ভিত্তির উপর দাঁড়িয়ে আমরা কম্পিউটারে কাজ করি।

একটু অতিসরলীকৃত একটা চেনা চেহারা দেওয়ার চেষ্টা করা যাক। ধরুন, আপনি একটা সেলফোন কানেকশন নিলেন, যা দিয়ে ইন্টারনেটে সংযোগ করা যায়। আপনি চালাতে পারলেন না আপনার গ্নু-লিনাক্স সিস্টেমে। লাগে মেল করলেন। ধরুন, জানলেন, যে ওই ধরনের ভৌত উপাদান এমনিতে কারনেলে দেওয়া নেই, কিন্তু ওয়াপ বা ওয়ারলেস-অ্যাকসেস-প্রোটোকল দিয়ে ইন্টারনেট করার জন্যে প্রয়োজনীয় মডিউল করে কারনেলে কম্পাইল করে

নেওয়া যায়। আপনার সিস্টেমে এই বাড়তি মডিউলটা দিয়ে আপনি কারনেল কম্পাইল করে নিলেন। কারনেলে এই বদল মাত্র আপনার সিস্টেমের কনফিগারেশনে এবং সিস্টেম প্রোগ্রামে কিছু প্রয়োজনীয় বদল ঘটে গেল। মানে মোট ওএসটাই বদলে গেল। এবার তার উপর দাঁড়িয়ে আপনি কানেক্ট করলেন নেটে এবং ব্রাউজার চালিয়ে নেট চরাতে শুরু করলেন। সেগুলো ইউজার প্রোগ্রাম।

২।। ইউনিক্স থেকে গ্নু-লিনাক্স

আগেই বললাম, ইউনিক্স সবচেয়ে বড় যে জায়গাটা নিয়ে এল — সেটা হল একই অপারেটিং সিস্টেমকে বারবার ব্যবহার করতে পারা, রিসাইক্লিং। হার্ডওয়ারের উপর নির্ভর করে থাকছে অপারেটিং সিস্টেমের একটা ছোট্ট জায়গা — যার নাম কারনেল। সিস্টেম আলাদা মানে আলাদা কারনেল, কারণ, এই কারনেলটা হল অপারেটিং সিস্টেমের ভিত। কিন্তু তার উপর যে ইমারতটা দাঁড়িয়ে আছে তাকে কিন্তু হার্ডওয়ার মাথায় রাখতে হয়না, মানে শেল, কমান্ড আর অ্যাপ্লিকেশনকে। যন্ত্রপাতির ডিটেইলস মাথায় রাখে কারনেল। ইউনিক্স-এর ক্ষেত্রে এই ইমারতটা লেখা সি-তে — একটা হাইলেভেল মনুষ্যবোধযোগ্য ভাষায়। তাই সেই অংশটাকে এবার যে কোনো মেশিনে নিয়ে যাওয়া যায়, শুধু ওই মেশিনের নিজস্ব কারনেলের ভিতটুকু লাগে। অর্থাৎ, এই প্রথম, বেল ল্যাবরেটরির থম্পসন এবং অন্যান্যদের কাজের ভিত্তিতে আমরা একটা অপারেটিং সিস্টেম পেলাম যা মেশিন-নিরপেক্ষ।

স্বাভাবিকভাবেই, সঙ্গে সঙ্গে উৎসাহিত হয়ে উঠল সফটওয়ার কোম্পানিগুলো। অ্যাদ্দিন তাদের প্রত্যেক মেশিনের জন্য আলাদা আলাদা সেট বানাতে হত, এবার তারা অন্তত আগের দশগুণ করে বেচতে পারবে এক একটা সফটওয়ার। অনেক নতুন কিছু এবার ঘটতে শুরু করল। আলাদা আলাদা বিক্রেতার আলাদা আলাদা মেশিন একই নেটওয়ার্কে এখন পরস্পর প্রতিপ্রাপকতায় আসতে পারছে, এর তথ্য ও চটকে দিচ্ছে, এর ছাপার কাজ আকার পাচ্ছে ওর প্রিন্টারে, ইত্যাদি। আগে যা সম্ভব ছিলনা, কারণ, তাদের অপারেটিং সিস্টেম ছিল আদ্যোপান্ত আলাদা, আর একটা কম্পিউটার কী করে অন্য কম্পিউটারের সঙ্গে কথা বলবে সেই নিয়মকানুনগুলো, যার টেকনিকাল নাম প্রোটোকল, সেই প্রোটোকলটাও তো মোট সিস্টেমেরই একটা অংশ। একজন ব্যবহারকারী এখন আলাদা আলাদা মেশিনে বসে কাজ করছে একই উপায়ে, একই ভাবে, একই কমান্ড দিয়ে — বা কমান্ডের বদলে মেশিনের একই জায়গায় একই লাথি মেরে — ব্রুস উইলিসের আরমাগেডনে স্পেসশিপের কম্পিউটারটা মনে করুন। ইন্ডিপেন্ডেন্স ডে-তেও, যতদূর মনে পড়ছে, জেফ গোল্ডব্লামের ভাইরাস কাজ শুরু করেনি মেশিন কিল খাওয়ার আগে অব্দি। এই পৃথিবীতো ব্যবসার, এখানে বিপ্লবী হতে গেলেও এনআরআইদের স্পনসরশিপ লাগে, এমতাবস্থায় সফটওয়ারওয়ালাদের এতাবৎ সমর্থন, ইউনিভার্সিটিবাজরা তো আগে থেকেই ছিল, জিতা রহো ইউনিক্স, ভীষণ জ্যান্ত রকমে বাড়তে বদলাতে নতুন নতুন ছানা পয়দা করতে লাগল ইউনিক্স, পরবর্তী দুটো দশক জুড়ে। দিনকে দিন আরো আরো নতুন কাজ করা যাচ্ছিল, নতুন ধরনের যন্ত্র, নতুন ছাঁদের প্রোগ্রাম, তাই হার্ড আর সফটওয়ারওয়ালারাও তাদের সিস্টেমে ইউনিক্স লাগিয়েই চলছিল।

পরপর মূল ইউনিক্স, এবং তার পিছনে পিছনে এর মূল ব্র্যান্ড বা ডিস্ট্রোগুলো কী ভাবে এসেছে তার একটা ছোট তালিকা। এখানে আমরা গুরুত্বপূর্ণগুলোই দেখিয়েছি কেবল। প্রথম দুটো ইউনিক্সের নামের বানানের তফাতটা খেয়াল করুন। পরে আসছি এই কথায়।

ইউনিক্সের সময়ধারা	
১৯৬৯	**UNICS**
১৯৭১	**UNIX** Time-Sharing System
১৯৭৪	MERT, PWB/UNIX
১৯৭৬	UNSW
১৯৭৭	LSX, Mini Unix, PWB, RT, TS, USG, Wollogong
১৯৭৮	1BSD, CB Unix, Interactive IS
১৯৭৯	2BSD, BRL Unix, UCLA Secure Unix, UNIX 32V
১৯৮০	3BSD, 4BSD, UCLA Locus, Xenix OS

১৯৮১	QUNIX, UNIX System III
১৯৮২	HP-UX, PC/IX, SPIX, SunOS, Ultrix-11, UNIX System IV, Venix
১৯৮৩	Coherent, mt Xinu, Sinix, UNIX System V
১৯৮৪	Dynix, **Minix**, QNX, SCO Xenix, Ultrix 32M, Unicos, UNIX System V Release 2, Xinu
১৯৮৫	IBM IX/370, Interactive 386/ix, Mach, MIPS OS, UNIX System V/286
১৯৮৬	A/UX, AIX/RT, Chorus, GNU, Plan 9, UNIX System V Release 3, UNIX System V/386
১৯৮৭	CTIX, HPBSD, SCO Xenix System V/386
১৯৮৮	BSD Net/1, Chorus/MiX, IBM AOS, IRIX, more/BSD, NeXTSTEP, Ultrix, UNIX System V Release 4
১৯৮৯	Acorn RISC iX, AIX PS/2, AIX/6000, Atari Unix, BOS
১৯৯০	AIX, AIX/370, AMiX, OSF/1, Solaris 1
১৯৯১	AIX/ESA, ASV, BSD Net/2, BSD/386, **Linux**, RISC iX, Trusted Xenix, UNIX Interactive
১৯৯২	386 BSD, AOS Reno, BSD/OS, Solaris 2
১৯৯৩	Dynix/ptx, FreeBSD, HP-UX BLS, MVS/ESA OpenEdition, NetBSD, Unicox-max, UnixWare
১৯৯৪	4.4BSD Lite 1, ArchBSD, Lites, Open Desktop, SCO UNIX
১৯৯৫	4.4BSD Lite 2, AOS Lite, DEC OSF/1 ACP, Digital Unix, OpenBSD, OpenServer 5, Sinix ReliantUnix, Trusted IRIX/B
১৯৯৬	Mk Linux, OPENSTEP, OS/390 OpenEdition, QNX/Neutrino, Unicos/mk
১৯৯৭	OS/390 Unix, ReliantUnix, Rhapsody
১৯৯৮	Monterey, Trusted Solaris, UnixWare 7, xMach
১৯৯৯	Darwin, Mac OS X, Mac OS X Server, Tru64 Unix
২০০০	AIX 5L, Debian GNU/Hurd, HP-UX 11i, Minix-VMD
২০০১	GNU-Darwin, Open UNIX 8, QNX RTOS, Security-Enhanced Linux, z/OS Unix System Services
২০০২	MicroBSD, MirBSD, SCO UnixWare 7
২০০৩	DragonFly BSD, ekkoBSD

তালিকার তথ্যসূত্র http://www.levenez.com/unix/, গিয়ে দেখুন, ভারি চমৎকার ছবি দিয়ে বোঝানো আছে বেল ল্যাবরেটরির মূলটা থেকে শুরু করে গোটা ইতিহাসটা।

ইউনিক্স প্রথমদিকে মিলতে মৈনাক মেইনফ্রেমে, বা তার ছোটভাই মিনি মেশিনে। ইউনিক্স চাও? কোনো বিশ্ববিদ্যালয়ে যাও, নিদেন কোনো সরকারী দপ্তরে, বা কোনো সুবৃহৎ ব্যবসায়। কিন্তু আজ আমরা যে পিসি মেশিনগুলোকে চিনি, তাদের আদিজনিতৃ মাইক্রো মেশিনেরা কিন্তু ইতিমধ্যে গোকুলে বাড়ছিল। আশির দশক তার প্রাত্যহিক পেইন্টের অনিবার্যতায় মানে প্রৌঢ়তায় পৌঁছতে পৌঁছতেই মাইক্রোরা ল্যান্ড করতে শুরু করল — এবারে আর শুধু বাজারে ব্যবসায় বা বিশ্ববিদ্যালয়ে না, ডাইরেক্ট টু হোম, হোম কম্পিউটারের ধারণা রৈ রৈ করে বাড়ছে। আর, ততক্ষণে, ইউনিক্স-ও পিছিয়ে নেই, বাজারে এসে গেল ইউনিক্স-এর হোম ভার্শন। কিন্তু কেন থম্পসনদের উন্মুক্ত আদি লিপির, ওপন সোর্স কোডের, দিন আর নেই, হোম মেশিনের উপযুক্ত প্রতিটি ইউনিক্সই ততদিনে কিনতে হচ্ছে।

লিনাস টরভাল্ডস তখন হেলসিঙ্কি বিশ্ববিদ্যালয়ে কম্পিউটার শাস্ত্রের ছাত্র। টরভাল্ডস-এর মনে হল, দাতব্য অ্যাকাডেমিক ইউনিক্স-এর একটা হোম সংস্করণ ফ্রিতে দেওয়া গেলে ভারি ভাল হত। আগেই বলেছি, মডেল তো তার কাছে ছিলই, ট্যানেনবম-এর মিনিক্স, কোড লেখা শুরু হল। শুধু লেখা না, রীতিমত একটা গবেষণা, যাতে অন্য অনেককে তার নিজের পাশে দরকার পড়ল, একা একা যে সমস্যাগুলোর সমাধান করা যাচ্ছিল না। এটা সেইরকম একটা আবেদন, নেটে, সাহায্যের আশায় — আমরা কম্প-অস-মিনিক্স নিউজ গ্রুপে টরভাল্ডস-এর মূল আবেদনটাই তুলে দিলাম। এর মধ্যে যে পোজিক্স স্ট্যান্ডার্ড এসেছে তা নিয়ে আমরা চার নম্বর দিনে কথা বলেছি। টরভাল্ডসের চেষ্টা ছিল একটা সর্বসাধারণের জন্য সম্পূর্ণ ফ্রি অপারেটিং সিস্টেম দেওয়া, যা মূল ইউনিক্সের সঙ্গে মিলবে। এই মেলার জায়গাটা রাখতেই তার পোজিক্স-এর কথা বলা। লিনাসের এই কারনেলটা বানাতে গিয়ে যে প্রাথমিক

সফটওয়ারগুলো প্রয়োজন পড়ল তা তদ্দিনে এসে গেছে, রিচার্ড স্টলম্যানের শুরু করা গ্নু তথা ফ্রি সফটওয়ার আন্দোলনের দৌলতে।

```
From: torvalds@claava.Helsinki.FI (Linus Benedict Torvalds)
Newsgroups: comp.os.minix
Subject: Gcc-1.40 and a posix-question
Message-ID: 1991Jul3.100050.9886@claava.Helsinki.FI
Date: 3 Jul 91 10:00:50 GMT
Hello netlanders,
Due to a project I'm working on (in minix), I'm interested in the posix
standard definition. Could somebody please point me to a (preferably)
machine-readable format of the latest posix rules? Ftp-sites would be
nice.
```

নতুন নতুন যন্ত্রাংশ বা গ্যাজেট নিয়ে আমরা এখন প্লাগ-অ্যান্ড-প্লে-র কথা বলি, যাকে অনেকে প্লাগ-অ্যান্ড-প্রে-ও বলেন, মেশিনে লাগাও তারপর বিশ্বকর্মা হোক, কি অন্য কোনো ফেভারিট ঈশ্বর, তার কাছে প্রার্থনা শুরু করো, ঠাকুর গ্যাজেটটা যেন কাজ করে, আরো যদি প্রতিবেশটা হয় উইন্ডোজ, অন্তত একবার রিস্টার্ট তো বাধ্যতামূলক। কিন্তু প্লাগ-অ্যান্ড-প্রে তখনো না-থাকুক রোজ রোজ গজাচ্ছিল নতুন নতুন যন্ত্রাংশ, সে নতুন ধরনের নতুন স্পিডের হার্ডড্রাইভ হোক, কিম্বা নতুন ধরনের ভিডিওকার্ড বা মনিটর যাই হোক। তার জন্যে নিত্য নতুন আলাদা আলাদা রকমের ড্রাইভার তো দিতে হবে, নইলে অপারেটিং সিস্টেম ওগুলোকে কাজে লাগাবে কী করে? যেটা ঘটছিল তা হল নতুন একটা যন্ত্রাংশ বাজারে এলে কেউ না কেউ সেটা গ্নু-লিনাক্স সিস্টেমে ব্যবহার করার চেষ্টা করছে এবং সেখান থেকে তৈরি হচ্ছে একটা নতুন ড্রাইভার। তদ্দিনে 'লিনাক্স' নামটাও চালু হয়ে গেছে, যদিও 'কোড লিনাক্স' নামে ফিল্ম মোতাবেক লিনাস প্রস্তাব করেছিলেন 'ফ্রিক্স' — ফ্রিক লোকেরা যা ব্যবহার করছে, নার্ড বা ফ্রিক নামদুটো তদ্দিনে লিনাক্সোৎসাহীদের সঙ্গে জুড়ে গেছে, আমরা যেমন পাগল বা খ্যাপা বলি। তথাগত এই পাঠমালার প্রথম খসড়া পড়ার পরে খেয়াল করিয়ে দিয়েছিল, এর সঙ্গে ফ্রিডম-এর 'ফ্রি'-টারও একটা মাত্রা আছে। জনপ্রিয় নাম লিনাক্স-এর জায়গায় কেন আমরা গ্নু-লিনাক্স ডাকছি, সেটা আরো স্পষ্ট হবে পাঠমালাটা পরে।

তাই নতুন নতুন আসা হার্ডওয়ারের জন্যে লেখা নতুন নতুন ডিভাইস ড্রাইভার গ্নু-লিনাক্স ভাণ্ডারে বেড়েই চলছিল। এই কোডলিখিয়েরা তাদের নিজের নিজের পিসিতেই থামেনি, ব্রহ্মার এই আড্ডায় প্রায় যা যা হার্ডওয়ার পাওয়া সম্ভব বা অসম্ভব তাদের সবারই কোড প্রায় লিখে ফেলা হয়েছে। এই হবিবাজ 'নার্ড' বা 'ফ্রিক'-দের কল্যাণে লিনাক্সনীয় হার্ডওয়ারের সংখ্যা বেড়েই চলেছে। এই খ্যাপাদের দৌলতেই শুধু যেকোনো পিসি মেশিনে অনাবিল চালানো যায় লিনাক্স তাই নয়, প্রচুর অদ্ভুত বিচিত্র খ্যাপাটে যন্ত্রাংশও চমৎকার কাজ করে লিনাক্সে — অন্য সিস্টেমে যাদের ফেলে দিতে হত।

টরভাল্ডস-এর ওই আবেদনের ঠিক দুবছর পরে লিনাক্স-ব্যবহারকারীর সংখ্যা দাঁড়াল বারো হাজার। শুরু হয়েছিল সাইবারখ্যাপাদের নিয়ে — এখন সুস্থিত সঙ্গত সংসারী মাথায়-তেল-মেখে-চান-করা ব্যবহারকারীর সংখ্যা ক্রমে বাড়ছিল। গোটা সময় জুড়ে একবারো পোজিক্স স্ট্যান্ডার্ড-এর নিয়মকানুন ভাঙা হয়নি। ইউনিক্স-এর প্রতিটি কাজ বা ক্রিয়া বা ফাংশন একে একে যোগ করা হচ্ছিল গ্নু-লিনাক্সে। ক্রমে গ্নু-লিনাক্স একটা পূর্ণাঙ্গ ইউনিক্স প্রতিরূপ বা ক্লোন হয়ে দাঁড়াল। যা ওয়ার্কস্টেশন থেকে শুরু করে উচ্চশক্তির সার্ভার অব্দি সবকিছুতে ব্যবহার করা যায়। আজ হার্ড এবং সফটওয়ার বাজারে প্রতিটি গুরুত্বপূর্ণ কোম্পানির একটা করে পুরোদস্তুর গ্নু-লিনাক্স টিম আছে — যারা নতুন নতুন কোড নির্মাণ করেই চলেছে।

ডেস্কটপ ছোট মেশিনের জগতেও লিনাক্স আজ পূর্ণবিকশিত। প্রথমদিকের লিনাক্স কোডলিখিয়েদের দৌলতে সবচেয়ে জরুরি ছিল নেটওয়ার্কিং ইত্যাদি কাজ। এখন অফিস স্যুট বলতে আমরা যা বুঝি সেখানেও লিনাক্স দিয়ে কাজ করা যায় অন্য যে কোনো অফিস স্যুটের সঙ্গে সমান রকমে। বহু ক্ষেত্রেই আরো অনেক ভালো করে। আমাদের লিনাক্সীদের এটা ভালো লাগুক বা না-লাগুক এই ডেস্কটপ জগতে মাইক্রোসফট এখনো সবচেয়ে বড় শক্তি, তার সঙ্গে প্রতিযোগিতায় একের পর এক নতুন নতুন ভালো ভালো সফটওয়ার গ্নু-লিনাক্সে রোজই যোগ হচ্ছে।

ওয়ার্কস্টেশনে কাজের উপযোগী করে। একটা সহজ গুই-নির্ভর ইন্টারফেস, উইন্ডোজ যেমন, এবং ওয়ার্ড একসেল পাওয়ারপয়েন্ট ইত্যাদি মাইক্রোসফটের অফিস প্রয়োগের গ্নু-লিনাক্স বিকল্প।

সার্ভারের জগতে গ্নু-লিনাক্স সবচেয়ে দৃঢ় এবং জোরালো সিস্টেমদের একটা। যেখানে ডেটাবেস এবং বাণিজ্যিক ব্যবহারের প্রতিটির জন্য নির্দিষ্ট গ্নু-লিনাক্স সমাধান আছে। ইন্টারনেটে প্রতিমুহূর্তে আমরা যার রেফারেন্স পাই সেই আমাজন বুক শপ চলে গ্নু-লিনাক্সে। আমেরিকার পোস্ট অফিস ব্যবস্থা, জার্মান সৈন্যবাহিনী চলে গ্নু-লিনাক্সে। কলকাতার সিইএসসির কাজের গ্নু-লিনাক্স ব্যবস্থায় কাজ করেছে এমন ছেলে আছে আমাদের কলকাতা লাগে। ইন্টারনেটে সার্ভিস প্রোভাইডাররা রোজ আরো আরো বেশি করে বেছে নিচ্ছে গ্নু-লিনাক্সকে। গ্নু-লিনাক্সে নেট-এর কাজ করাও সবচেয়ে নির্ঝঞ্ঝাট। টাইটানিক সিনেমা বানানোর গোটা সিস্টেমটা চালানো হয়েছিল লিনাক্সে।

অন্যদিকে স্মার্টকার্ড, মোবাইল ফোন, পিডিএ (PDA — **P**ersonal-**D**igital-**A**ssistant — বলতে এক কথায় বোঝায় হালকা হাতের তালুতে ধরে যাওয়ার মত পামটপ, নির্দিষ্ট কিছু কাজ করে, সময়তারিখ রাখা, নোট নেওয়া, ছোট ডেটাবেস, ক্যালকুলেটর ইত্যাদি, খুব দামি গুলোতে মাল্টিমিডিয়া অব্দি হয়) বা অর্গানাইজার ইত্যাদি এমবেডেড সিস্টেমে, মানে যেখানে অপারেটিং সিস্টেমটা ইতিমধ্যেই সদাসর্বদাই, অলওয়েজ অলরেডি, হার্ডওয়ারের মধ্যে রয়েছে — এই জগতটায় লিনাক্স ক্রমে প্রায় সর্বব্যাপী হয়ে উঠছে। কী কী ধরনের হার্ডওয়ারে একটা অপারেটিং সিস্টেম চালানো যাবে সেই নিরিখে গ্নু-লিনাক্স-এর পাশে আর কোনো অপারেটিং সিস্টেমই আসতে পারেনা। এর ইতিহাসটা তো আমরা আগেই বলেছি। অর্থাৎ, গ্নু-লিনাক্স এখন একটা পূর্ণাঙ্গ ব্যবস্থা — আমাদের সবার প্রয়োজনই যা থেকে মিটছে। প্রতিটা নতুন দিন নতুন প্রয়োজন গজিয়ে উঠবে, জ্যান্ত লিনাক্সীদের নিয়ে জ্যান্ত গ্নু-লিনাক্সও বদলাতে থাকবে।

৩।। ইউনিক্স থেকে গ্নু-লিনাক্স — সালগত ক্রমপঞ্জী

এই আলোচনাটা আমরা করব একদম একটা ইতিহাস বইয়ের শেষে দেওয়া সালপঞ্জীর মত করে, সাল ধরে ধরে। তার মধ্যে দু-এক জায়গায় দু-একটা কথা বাড়তি বলে নেব ধরিয়ে দেওয়ার জন্যে। এই পঞ্জীটা বানাতে গিয়ে আমি মূলত চারটে ওয়েবসাইট থেকে সাহায্য পেয়েছি। এক, বাংলাদেশ লাগ-এর (http://bdlug.hypermart.net/) জন্যে রাজিব হাসানের লেখা 'হিস্ট্রি অফ লিনাক্স'। দুই, ইউনিক্স ইতিহাসের সাইট http://www.levenez.com/unix/। তিন, লিনাক্সের ইতিহাস নিয়ে পিটার সেলাসের বক্তৃতা http://technetcast.ddj.com/ থেকে পাওয়া এক ঘন্টার এমপিথ্রি ফাইল — বক্তৃতাটা ভারি মজার, জিএলটির রিসোর্স সিডিতে দেওয়া আছে, সতেরো মেগাবাইট সাইজ। চার, বেল ল্যাবরেটরির নিজের ইউনিক্স ইতিহাসের সাইট, http://www.bell-labs.com/history/unix/। পঞ্জীটা শুরু হচ্ছে ইউনিক্সেরও জন্মের আগে, ১৯৫৭ থেকে।

১৯৫৭।। বেল ল্যাবরেটরিতে নিজেদের গবেষণার কাজে তখন ব্যাচ সিস্টেম ব্যবহৃত হচ্ছিল, চার নম্বর দিনের জেনারেশনের ইতিহাসের সঙ্গে মিলিয়ে নিন। এই ব্যাচ সিস্টেমে কাজের প্রক্রিয়াটা কখনোই খুব আরামপ্রদ হচ্ছিল না। বহু কিছু কাজ নিজেদের হাতে করে, প্রত্যেকবার, করে দিতে হচ্ছিল। তৈরি হল বেসিস (BESYS) বলে একটা অপারেটিং সিস্টেম। এগুলো সবই ওএস-এর প্রাগৈতিহাসিক সংস্করণ। এখনকার ওএস-দের সঙ্গে একে মেলানো অনেকটা ছিপকলির সঙ্গে ডাইনোসর মেলানোর মত। কিছুটা মিল বলা যায় চার নম্বর দিনে আমাদের দেখানো ফোর্ট্রান মনিটরিং সিস্টেমের সঙ্গে।

১৯৬৫।। বেল ল্যাবরেটরি তখন ক্রমে তৃতীয় জেনারেশন কম্পিউটারে চলে যাচ্ছে, বড় একটা ওএস তাদের আশু প্রয়োজন। জেনেরাল ইলেকট্রিক এবং ম্যাসাচুসেটস ইনস্টিটিউট অফ টেকনোলজির সঙ্গে মিলে একটা যৌথ প্রকল্পে হাত দিল বেল ল্যাবরেটরি। আমরা পেলাম সেই সকল টার্মিনালের এক কেন্দ্র, গোটা শহরের এক সিস্টেম — সেই সর্বশক্তিমান সর্বত্রগামী মাল্টিক্স (MULTICS)।

১৯৬৯।। উনসত্তরে ঘটল চার চারটে ঘটনা, সেই মুহূর্তে তাদের একটার সঙ্গে আর একটাকে মেলানো নোস্ত্রদার বাবার দ্বারাও সম্ভব ছিলনা। কিন্তু পরে, ইতিহাসের পাতায় তারা মিলে গেল একই ঘটমানতায়। পিটার সেলাস-এর মত করে বলতে গেলে, যা শুধু কম্পিউটার না, পৃথিবীর বিজ্ঞান ও প্রকৌশলের মানচিত্রটাকেই

বদলে দিল। (১) মানুষ চাঁদে পা দিল। (২) অগাস্ট সেপ্টেম্বরে কেন থম্পসন আর ডেনিস রিচি তৈরি করলেন ইউনিক্স-এর প্রথম কাঠামো। তার প্রথম এবং আশু কারণ, এটিঅ্যান্ডটি (AT&T) মাল্টিক্স ব্যবহার বন্ধ করে দিল, এবং থম্পসন আর রিচির দরকার পড়ল একটা নতুন অপারেটিং সিস্টেমের দরকার পড়ল, যে সিস্টেমে তারা তাদের প্রিয় গেমস, 'স্পেস ট্রাভেল' খেলে উঠতে পারবে। এই খেলাটা তারা খেলত, ডিইসি বা ডেক-এর পিডিপি ৭ মেশিনে। পিডিপি সিরিজের কথা মনে পড়ছে? (৩) চৌঠা ডিসেম্বর একই সাথে বসল চারটে ইম্প (IMP — **I**nterface-**M**essage-**P**rocessors) বা আদিমতম প্রোটোটাইপ রুটার, অর্থাৎ এককথায় ইন্টারনেটের অঙ্কুরোদগম। এই চারটে জায়গা হল, এলএ, সান্টা-বারবারা, মেনলো পার্ক এবং উটা বিশ্ববিদ্যালয়। (৪) ২৮শে ডিসেম্বর টরভাল্ডস পরিবারে জন্মালো নতুন বাচ্চা, নাম রাখা হল লিনাস।

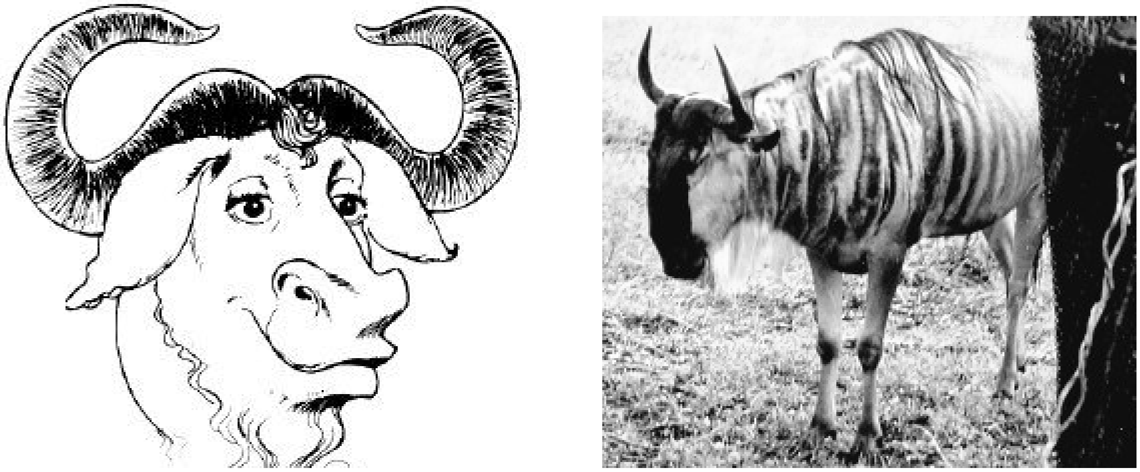
১৯৭১।। ইউনিক্স প্রথম এডিশন প্রকাশিত হল। এই প্রথম তার বানান হল UNIX। খোজা করা মাল্টিক্স — এই প্যারডি থেকে একটা পূর্ণবিকশিত ব্র্যান্ডনেম তৈরি হল, ওএস-এর। এই ইউনিক্সের ম্যানুয়ালের ভূমিকায় কেন থম্পসন আর ডেনিস রিচি লিখলেন একটা তাৎপর্যপূর্ণ বাক্যাংশ, ' … there are now only ten installations of UNIX os, but more to come …', '...এই মুহূর্তে ইউনিক্স অপারেটিং সিস্টেম ব্যবহার হচ্ছে মাত্র দশ জায়গায়, কিন্তু আরো আসছে ...'। এই 'আরো আসছে' মানে ঠিক কতটা আসছে তা যদিও, বোধহয়, ওরা নিজেরাও আন্দাজ করতে পারেননি। এই ইউনিক্সে ছিল মোট ষাটটা কমান্ড। এর পর, আপনারা যখন এই কমান্ডগুলোর আজকের চেহারার সঙ্গে পরিচিত হতে থাকবেন, ছয় সাত আট নম্বর দিনে, এই পুরোনো চেহারাগুলোর সঙ্গে মিলিয়ে যাতে আমোদ পেতে পারেন, তাই কয়েকটা কমান্ডের উদাহরণ দিলাম এখানে। যেমন, '`b`', মানে বি (B) ভাষায় লেখা প্রোগ্রাম কম্পাইল করো। বি ভাষার কথা তো আমরা উল্লেখ করেছি আগেই, সি নামের কম্পিউটার ভাষার পূর্বসূরী। সি (C) ভাষায় লেখা প্রোগ্রাম কম্পাইল করার জন্যে জিসিসি-কে দেওয়া কমান্ডটা মনে করতে পারছেন? আর একটা কমান্ড ছিল '`boot`', মানে মেশিন রিবুট করো, অফ করে অন করো। বা কনক্যাটেনেট করার, ফাইল স্ক্রিনে দেখানোর এবং পরপর জুড়ে দেওয়ার কমান্ড '`cat`', বা, যে ডাইরেক্টরিতে কাজ করছি সেটা বদলানোর কমান্ড '`chdir`'। বা '`cp`' — ফাইল কপি করো, '`chmod`' — ফাইলের ব্যবহারবিধি বদলাও, '`chown`' — ফাইলের মালিকানা বদলাও, '`mv`' — ফাইলের জায়গা বদলাও বা নাম বদলাও, '`wc`' — ওয়ার্ড-কাউন্ট বা শব্দসংখ্যা গোনো, বা '`who`' — কে বা কে-কে এই মুহূর্তে সিস্টেম ব্যবহার করছে সেটা দেখাও। ইত্যাদি। এই তালিকাটা লিখতে গিয়ে আমার নিজের যে কী বিচিত্র লাগছে, এর একদম প্রথমটা বাদ দিয়ে অন্য সবকটাই রোজ ব্যবহার করি, একদম রোজ, অথচ এগুলো আজ থেকে ঠিক পঁয়ত্রিশ বছর আগে থম্পসন আর রিচির বানানো। একাত্তরের এই সিস্টেমে পরবর্তীকালের অন্য ইউনিক্সগুলোর, যার মধ্যে ওপন-সোর্স ইউনিক্স বা গ্নু-লিনাক্সও একটা, সবচেয়ে গুরুত্বপূর্ণ জায়গার একটা একেবারেই নেই — তা হল পাইপ, যার বিশদ আলোচনা আসবে দশ নম্বর দিনে, আমাদের পাঠমালার এগারো নম্বর অংশে।

১৯৭২।। ইউনিক্স-এর দ্বিতীয় ভার্শন বেরোলো। আর সবচেয়ে বড় যেটা, ডেনিস রিচি তার বি-গত বিরক্তি থেকে সি-তে পৌঁছলেন। বি ভাষায় কাজের অসুবিধেগুলো একটা নতুন কেজো এবং ভালো কম্পিউটার ভাষার আবির্ভাবকে জরুরি করে তুলেছিল অনেকের কাছেই।

১৯৭৪।। বিল জয় এসে ইউনিক্স-এর কাজে যোগ দিলেন, এড (ed) বলে পুরোনো এডিটর-টা ব্যবহার করতে করতে তিতিবিরক্ত জয় বানালেন এক্স (ex) বলে একটা এডিটর। অপারেটিং সিস্টেম বা প্রোগ্রামিং-এর কাজে কমান্ড এডিটর কতটা গুরুত্বপূর্ণ সেটা তো এক নম্বর দিনেই আমরা আলোচনা করেছি। এবং ছবি নিয়ে খেলা করা বা গুই মোড আসতে তখনো বহু বহু দেরি। তখন এডিটরের ভূমিকা ছিল আরো অনেক বড়। এই এড কিন্তু খুবই শক্তিশালী একটা প্রোগ্রাম। এবং পরবর্তী প্রচুর এডিটরই পিছনে এডকে ব্যবহার করে। এডের একটা ভারি চমৎকার টিউটোরিয়াল আছে কারনিঘ্যান পাইকের বইয়ের অ্যাপেনডিক্স-এ। এই এড-স্রষ্টা বিল জয়ই পরে বানান ভিআই (vi) বলে একটা এডিটর যা গোটা ইউনিক্স জগতের কমান্ড এডিটিং-এ প্রায় একচ্ছত্র ছিল, রিচার্ড স্টলম্যানের ইম্যাক্স আসার আগে অব্দি।

১৯৭৫।। ইউনিক্স ভার্শন ছয় বেরোলো। এবং আরএমএস বা রিচার্ড স্টলম্যান লিখলেন তার ইম্যাক্স-এর প্রথম ভার্শন। রিচার্ড স্টলম্যান ছিলেন এমআইটির বিখ্যাত আর্টিফিশিয়াল ইন্টেলিজেন্স ল্যাবরেটরির সাইবারবিজ্ঞানী। ওই আর্টিফিশিয়াল ইন্টেলিজেন্স ল্যাবের অন্য প্রায় প্রত্যেকেই যখন চড়া দামে বিক্রি হয়ে যাচ্ছে বড় বড় বাণিজ্যিক সংস্থায়, ঠিক তখনই স্টলম্যান শুরু করেন তার ফ্রি সফটওয়ার নিয়ে লড়াই। সময়টা খেয়াল করুন, পঁচাত্তর সাল। ইন্টারনেট তখনো আসেনি। ইন্টারনেটের পূর্বপুরুষ ছিল আর্পানেট, সেই আর্পানেটে সাইটের সংখ্যা তখন একশোরও নিচে, সাইট মানে আজকে আমরা যা বুঝি তার থেকে বহু যোজন দূরে তারা, ইন্টারনেট ঘটে উঠতে তখনো অনেক দেরি, যে ইন্টারনেটকে বাদ দিয়ে, আরো ষোল বছর পরে গ্নু-লিনাক্স ব্যাপারটা ঘটেই উঠতে পারত না। একটু বাদেই আমরা আসছি সেই প্রসঙ্গে। এই আর্পানেট (ARPA — **A**dvanced-**R**esearch-**P**rojects-**A**gency-NET) শুরু হয়েছিল উনিশশো ষাট-এর দশকে আমেরিকার ডিপার্টমেন্ট অফ ডিফেন্সের অ্যাডভান্সড রিসার্চ প্রোজেক্টের কাজে, যাতে বিভিন্ন বিশ্ববিদ্যালয় এবং গবেষণা সংস্থাগুলোর মধ্যে সরাসরি তথ্য বা ইনফরমেশন যাতায়াত ঘটতে পারে, পরে সত্তরের দশকে এর নাম হয় ডার্পা। এই আর্পা বা ডার্পা প্রাথমিকভাবে ব্যবহার হত মিলিটারির কাজে, যার থেকে পরে ইন্টারনেট গজিয়েছে। এই বিবর্তনটার একটা বেশ ইন্টারেস্টিং ইতিহাস আছে। নেটওয়ার্কিং বা বিশেষ করে টিসিপি/আইপি প্রোটোকলের ইতিহাসের এই অংশটা। কিন্তু সেটা আমাদের এই পাঠমালার এক্তিয়ারের বাইরে।

১৯৭৬।। ছিয়াত্তর সালে বলার মত, বা, আসলে না-বলার মত, ঘটনা একটাই। বিল গেটস-এর একটা চিঠি। ‘ওপেন লেটার টু দি হবিস্টস’। যাকে সেলাস তার ওই ঘন্টাভর বক্তৃতায় ডেকেছেন ‘দি ইনসিপিয়েন্ট ফুটপ্রিন্ট অফ দি র‌্যাপ্টর অফ রেডমন্ড বলে’। রেডমন্ড হল গেটস-এর ঠিকানা, আর র‌্যাপ্টরের মূল অর্থ শিকারী মাংসভুক পাখি, কিন্তু জুরাসিক পার্কে এতবার র‌্যাপ্টর ব্যবহার হয়েছে ভেলোসির‌্যাপ্টর অর্থে, সেই ঘিনঘিনে ছোট সাইজের ডাইনোসরগুলোর নাম হিশেবে, ওই উদ্বৃত্ত অর্থটা কিছুতেই এড়ানো যায়না এই দু-হাজার তিনের পৃথিবীতে। এই চিঠিতে বিল গেটস হবিস্টদের বা হবি-বাজদের উদ্দেশ্য করে লিখেছিলেন, ‘...most of you steal your software...’ — তোমরা বেশিরভাগই সফটওয়ার চুরি করো। এখানে ‘হবি’ শব্দটার এই ব্যবহারটা বেজায় আমেরিকান, আমাদের অনেকটাই অপরিচিত। এটা বলতে বোঝায় ওই কম্পিউটার হুইজ কিডদের। বিল গেটসের চিঠির প্রায় পনেরো বছর পরে, ১৯৯১-এ, কম্প-অস-মিনিক্স মানে মিনিক্সের নিউজগ্রুপে লিনাক্স যখন প্রথম তার বানানো কারনেলটার খবর দেয়, একুশ বছরের সেই সদ্যযুবক, তাতে লেখে, I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu) for 386(486) AT clones, আমি একটা ফ্রি অপারেটিং সিস্টেম বানাচ্ছি। নিছক একটা হবি, শখ — গ্নু, মানে রিচার্ড স্টলম্যানের গ্নু-র মত বড় বা পেশাদার কিছু নয়। এই চিঠির ‘হবি’ শব্দটার সঙ্গে মেলান গেটস-এর চিঠিটাকে। এই ‘হবিস্ট’ শব্দটার ধনাত্মক উদ্বৃত্ত অর্থটা পরে গোটাটাই স্থানান্তরিত হয়ে গেছিল ‘হ্যাকার’ (hacker) শব্দটায়। এখানে একটা কথা খেয়াল করিয়ে দেওয়া ভালো, ‘হ্যাকার’ বা ‘হ্যাক করা’ শব্দবন্ধ, মূলত অর্ধসাক্ষর সাংবাদিকদের কল্যাণে, দাঁড়িয়ে গেছে একটা খারাপ বিশেষণ — যারা অন্যের মেশিনে অন্যের নিজস্ব জগতে ঢুকে পড়ে এবং নিজের লাভের বা মজার জন্যে নানা বেআইনি কাজ করে। কিন্তু, কম্পিউটার লেখালেখির জগতে পড়ার সময় খেয়াল রাখবেন, ‘হ্যাকার’ শব্দটা রীতিমত সম্মানিত শব্দ, যারা কম্পিউটারের কার্যধারা একদম ভিতর থেকে বোঝে এবং বদলাতে পারে, মানে সিস্টেম বিষয়ে যথেষ্ট পারদর্শী প্রোগ্রামার। ওই খারাপ অর্থটার জন্যে সঠিক শব্দটা হল ‘ক্র্যাকার’ (cracker)। ‘হ্যাকার’ শব্দটার অর্থ এই একশো আশি ডিগ্রি ঘুরে যাওয়ার ভিতরে একটা গোপন ক্রিয়া আছে। আজকের পৃথিবীর সবচেয়ে সমর্থ এবং কুলীন ব্যবসা কম্পিউটার, আর সব কিছুকে যা শাসন করে। এই সাইবার শিল্পের পুঁজির আজ অব্দি সবচেয়ে বড় আতঙ্কটা তৈরি করেছিল হ্যাকাররা, এদের সাজানো বাগানে একপাল গ্নু ছেড়ে দিয়ে। ফ্রি লিবর ওপন — যে চেহারাতেই আসুক, সাইবার-বিজ্ঞানে উণ্মুক্ততা এনেছে এই হ্যাকাররাই। হ্যাকারদের প্রতি পুঁজির অবশ্যম্ভাবী আতঙ্ক এবং বিতৃষ্ণাটা এবার সংবাদমাধ্যম মারফত ছড়িয়ে গেছিল জনমানসে। ঠিক যে ভাবে শাসকের চিন্তাকে বারবার শাসিতের চিন্তা বানানো হয়েছে ইতিহাস জুড়ে।



গ্নু — ফ্রি সফটওয়ার ফাউন্ডেশনের ছবিতে এবং জঙ্গলে

১৯৭৭।। সাতাত্তরে এল বিএসডি প্রথম ভার্শন। এর সঙ্গে ছিল ওই এক্স এডিটর এবং পাস্কাল বলে একটা কম্পিউটার ভাষার বার্কলে সংস্করণ। এর জন্যে মোট দাম ধার্য ছিল কুড়ি ডলার। এই কুড়ি সংখ্যাটাকে মনে রাখুন, এখুনি আমাদের তুলনা করতে কাজে লাগবে। আর দুতিন বছরের মধ্যেই, শুধু যে এই দামটাই বীভৎস ভাবে বদলে যাবে তা নয়, এর সঙ্গে চালু হবে লাইসেন্সিং নিয়ে উন্মত্ত কড়াকড়ি। এই সময় অব্দি আবহাওয়াটা অনেকটাই বিদ্যাচর্চার, দাম যা নেওয়া হচ্ছে সেটাও ন্যূনতম যতটুকু প্রয়োজনীয়। এর পরেই আবহাওয়াটা চলে যাবে একচেটিয়া পুঁজির মুনাফাবাজিতে। আর সফটওয়ারের উপরেও আসবে বাধানিষেধ। যাতে একটা যুগের প্রোগ্রামারদের কাজের উত্তরাধিকার পরের যুগে আর পৌঁছবে না, পুঁজি তাকে আটবে দেবে। তার মুনাফার স্বার্থে। এতকাল অব্দি মানুষের বিদ্যাচর্চার বিকাশটা ছিল সামগ্রিক, প্রতিটা প্রজন্ম যা করত সেটা পেয়ে যেত পরের প্রজন্ম। সেই বিবর্তন প্রবাহটাকে আটকে দেওয়া হবে। আর সেই আটকে দেওয়াটার প্রভাব আর শুধু কম্পিউটারেই সীমাবদ্ধ থাকবে না, কারণ, আর মাত্র একটা দশকের মধ্যে এমন সময় চলে আসছে যেখানে বিদ্যাচর্চার প্রতিটি এলাকারই জীয়নকাঠি আসলে থাকবে কম্পিউটারের মধ্যে।

১৯৭৯।। চার নম্বর দিনে অনেকটা এবং আজকের আলোচনার শুরুতেও অপারেটিং সিস্টেম বা ওএস-এর যে স্থানান্তরযোগ্যতা বা পোর্টেবিলিটি নিয়ে কথা বলেছি আমরা — সেটা প্রথম বাস্তব হয়ে উঠল উনআশিতে এসে। বেল ল্যাবরেটরি ইউনিক্স ভার্শন সাত। এই গ্রহের প্রথম ওএস যা একাধিক মেশিনে চলল, ডেক-এর পিডিপি সিরিজ এবং ইন্টারডেটার একাধিক মডেলে। আর এই উনআশিতেই, ইউজনিক্স বা ইউনিক্স ইউজার গ্রুপের মিটিং-এ একটা ঘটনা ঘটল যা কয়েক বছরের মধ্যে বদলে দেবে গোটা সফটওয়ার তথা কম্পিউটারের ইতিহাসকে। টরন্টোয় ইউজনিক্স-এর মিটিঙে এটিঅ্যান্ডটি ঘোষণা করল, এখন থেকে তাদের ইউনিক্স-এর দাম পড়বে বিদ্যাচর্চার প্রতিষ্ঠানগুলোয় সিপিইউ পিছু ৭৫০০ ডলার করে, এবং পূর্ণাঙ্গ ব্যবসায়িক লাইসেন্সের জন্যে ষাট হাজার ডলার। মানে আগের প্যারার ওই কুড়ি ডলারের মাত্র তিরিশ হাজার গুণ। এই মিটিঙেই উপস্থিত ছিলেন অধ্যাপক অ্যান্ড্রু ট্যানেনবম, তার প্ল্যান ছিল তার আসছে বছরের ওএস-এর ক্লাসে ছেলেদের ইউনিক্স দিয়ে কাজ করানোর। তার মানে তার, তিরিশ জন ছাত্রের তিরিশটা মেশিনে ইউনিক্স লাগাতে গেলে লাগবে সাড়ে সাত হাজার করে মোট দু লক্ষ পঁচিশ হাজার ডলার মাত্র। ট্যানেনবম তার মিনিক্স সংক্রান্ত কাজ শুরু করলেন, এটিঅ্যান্ডটি-র এই উন্মত্ত লাইসেন্সিং-এর বিপরীতে — ছাত্রদের পড়াতে গেলে না করে কোনো উপায় ছিলনা।

১৯৮০।। আশিতে বেরোলো বিএসডির ভার্শন চার। এর পর থেকে নববই অব্দি চলল বিএসডি-র ভার্শন ৪.১, ৪.১এ, ৪.১বি, ৪.১সি, ৪.২, ৪.২, ৪.৩, ৪.৪ — এই বিদঘুটে নামের পরপর সংস্করণের শোভাযাত্রা। এর কারণও আর কিছু না, সেই এটিঅ্যান্ডটি। এটিঅ্যান্ডটি বার্কলে সফটওয়ারকে অনুমতি দিয়েছিল যে তারা অ্যাকাডেমিক গবেষণার জন্যে ভার্শন চার দিতে পারে, কিন্তু সেটাকে বদলে কোনো নতুন সংস্করণ বাজারে

ছাড়তে পারবে না। তাই বার্কলেকে প্রমাণ করতেই হচ্ছিল যে ওগুলো কোনো নতুন ভার্শন না, জাস্ট একটু আপডেট, ছাঁটকাট। সচরাচর মূল সংখ্যাটার বদল দিয়ে সংস্করণ বদলানো বোঝানো হয়, আর তার সংযোজন সংখ্যা দিয়ে খুচরো বদলগুলোকে।

১৯৮৪।। চুরাশির জানুয়ারিতে আরএমএস তৈরি করলেন তার গ্নু (GNU) প্রোজেক্ট। এবং এই চুরাশিতেই এল গ্নু-ভার্শন বা নতুন সংস্করণ ইম্যাক্স যা আগেরটার চেয়ে আমূল পরিবর্তিত। এবং ইম্যাক্স নিয়ে কিছু বলার লোভ সামলানো শক্ত, কিন্তু তার চেয়েও শক্ত কিছু বলা। সবচেয়ে সহজে যা বলা যায় তা হল, আসুন, গ্নু-লিনাক্স শিখুন। ইম্যাক্স সম্পর্কে একটা চ্যাংড়ামি বরং উল্লেখ করা যাক — নিন্দুকেরা বলে থাকে ইম্যাক্স নামের কমান্ড এডিটরটা আসলে গোপনে গোপনে একটা অপারেটিং সিস্টেম, যাতে খুব সহজসাধ্য একটা কমান্ড এডিটরের মৃদু অভাব আছে।

১৯৮৫।। গ্নুর ম্যানিফেস্টো বা ইশতেহার প্রকাশিত হল ডক্টর ডবস জার্নালে, মে পঁচাশি ইশুতে। এর আগে তিরাশি থেকেই এই ইশতেহারটা লিখছিলেন আরএমএস। যেখানে একটা ফ্রি অপারেটিং সিস্টেমের তৈরির কথা ঘোষণা করা হয়, এবং সেই অপারেটিং সিস্টেমের কাজের জন্যে প্রয়োজনীয় কম্পাইলার ইত্যাদি যন্ত্রপাতি তৈরির প্রয়োজনীয়তার কথাও। পঁচাশির এই লাইসেন্স প্রপীড়িত ইউনিক্স জগতে একটা নাটকীয় বদল আনলো গ্নু-র সি কম্পাইলার কালেকশন বা জিসিসি। যে কোনো ইউনিক্স ব্যবহারকারীর একটা প্রাথমিকতম প্রয়োজন ছিল একটা ভালো সি কম্পাইলার। এবং ঠিক এর আগে আগেই সান মাইক্রোসিস্টেম তার সফটওয়ার বিক্রির রকমটা বদলে ফেলেছিল। আগেই বললাম, ক্রমে নিয়ন্ত্রণটা বিদ্যাচর্চার জগত থেকে চলে যাচ্ছিল পুঁজির হাতে। সান মাইক্রোসিস্টেম তার সমস্ত সফটওয়ার এতদিন একসঙ্গে বেচত, একটা প্যাকেজ হিশেবে। এবার, আরো লাভের আশায়, আলাদা আলাদা ভাবে বিক্রি করতে শুরু করল। এবং, আলাদা ভাবে সি কম্পাইলারের প্যাকেজের জন্যে ধার্য করল একটা কদর্য রকমের চড়া দাম। ফ্রি সফটওয়ার ফাউন্ডেশন, যে নামটা তখনো আসেনি, কিন্তু স্টলম্যানের নেতৃত্বে আন্দোলনটা শুরু হয়ে গেছে, সেই এফএসএফের পালে একটা বড় হাওয়া এল এই জিসিসি দিয়ে। এই পঁচাশিতেই এল জিএনইউ পাবলিক লাইসেন্স বা জিপিএলের ভার্শন এক, আমরা আগের দিন যার আলোচনা করেছি। এখানে সেলাসের একটা খোঁচালো মন্তব্য না-দিয়ে পারছি না। সেলাস নাকি একাধিকবার আরএমএসকে মানে স্টলম্যানকে এই মন্তব্যটা শুনিয়েছে, এবং প্রত্যেকবারই স্বভাবসিরিয়াস আরএমএস এতে ব্যাপক খার খেয়ে গেছে। মন্তব্যটা এই যে, ফ্রি সফটওয়ার ফাউন্ডেশনের পুরষ্কারটা পাওয়া উচিত বিল গেটস-এর, কারণ তার এবং তার জাতীয় র‍্যাপ্টরদের উজবুক লোভ এবং যুক্তিহীন লাইসেন্সিং ছাড়া ফ্রি সফটওয়ার আন্দোলন কোনোদিন তৈরিই হতে পারত না।

১৯৮৬।। এল মিনিক্স।

১৯৮৭।। প্রেন্টিস হল ট্যানেনবম-এর ‘অপারেটিং সিস্টেম’ বইটা ছাপল, যার সঙ্গে এমনিতেই পাওয়া যাচ্ছিল ওই বারো হাজার লাইন কোড, যা দিয়ে তৈরি করে নেওয়া যায় একটা অপারেটিং সিস্টেম। এখন নেট থেকেও পাওয়া যায় মিনিক্স-এর এই কোড। কিন্তু তখনো নেট নেট হয়ে ওঠেনি, আগেই বলেছি।

১৯৯০।। আইবিএম তৈরি করল আইক্স (AIX — **A**dvanced-**I**nteractive-e**X**ecutive), ইউনিক্স-এর আইবিএম সংস্করণ।

১৯৯১।। কম্প-অস-মিনিক্সে লিনাস টরভাল্ডজ-এর সেই ছোট্ট বিনয়ী একটা চিঠি, দুনিয়ার কম্পিউটিং ইতিহাসকেই যা বদলে দিল। আর, খেয়াল করুন, নেট ইতিমধ্যেই নেট হতে শুরু করেছে। অন্তত এর মধ্যেই সেই জায়গাটা তৈরি হয়ে গেছে নেট-এর দৌলতে, যাতে টরভাল্ডজ পোস্ট করা মাত্র তা পলক ফেলতে না-ফেলতেই পৌঁছে যায় অজস্র মানুষের কাছে। পঁচাত্তরে ছিল একশোর নিচে, অগাস্ট ১৯৮১-তে নেট-এ সাইটের সংখ্যা দাঁড়ায় ২১৩, ১৯৯৪-এ যা ৭৫ লাখে পৌঁছে গেছিল। ৯১-এ লিনাস টরভাল্ডজ যখন তার মেল পাঠাচ্ছেন, তখন সাইটের সংখ্যা কম করে পাঁচ লাখ, যাদের কাছে সেই মুহূর্তে চলে যাচ্ছে সেটা। এই অবস্থাটা নিয়ে ভারি মজার কিছু উপাখ্যান আছে ‘কোড লিনাক্স’-এ। এই ৯১-এর ১৯শে ডিসেম্বর চালু হয়

প্রথম লিনাক্স নিউজগ্রুপ, কম্প-অস-লিনাক্স। জানুয়ারি বিরানববই-এ, যে সময়টা আমার নিজের জীবনে খুব গুরুত্বপূর্ণ, মানে যার কখনো পড়ার কথা মাথায় রেখেও এই পাঠমালাটা লিখছি আমি, সে যখন জন্মাচ্ছে, তখন লিনাক্স ব্যবহারকারীর সংখ্যা ঠিক একশোর উপর। ৯১-এর শেষে এসে লিনাক্স সংস্করণ ০.১০ ঘোষিত হল। তখনো এই কুচো অপারেটিং সিস্টেম দিয়ে কাজ করা যেত শুধু এটি হার্ড ডিস্কে, লগ-ইন বলে কিছু ছিলনা, সরাসরি ব্যাশ শেলে ঢুকে যেত। এর মানে কী সেটা আজই দেখব আমরা। ভার্শন ০.১১-এ এসে অবস্থা অনেকটা বদলেছিল, তাতে একাধিক ভাষার কিবোর্ড লাগানো গেল, ফ্লপি ডিস্ক ব্যবহার করা গেল, ভিজিএ ড্রাইভার এল।

১৯৯২।। বিরানববই-এর উনত্রিশে জানুয়ারি লিনাক্স বিষয়ে আপত্তি এল খোদ অ্যান্ড্রু ট্যানেনবম-এর কাছ থেকে, সেই কম্প.অস.মিনিক্স-এ, বিষয়, লিনাক্স চলেনা, লিনাক্স ইজ অবসলিট। “LINUX … is a giant step back … like taking an existing, working C program and rewriting it in BASIC …”, একটা চালু সি প্রোগ্রামকে ফের ওই বেসিকে লেখার মত — লিনাক্স হল কম্পিউটার মননকে পিছিয়ে নিয়ে যাওয়া। এর মধ্যের উপ্ত গোটা অপমানটাকে বোঝাই সম্ভব না, একজন সিরিয়াস প্রোগ্রামসচেতন মানুষের কাছে বেসিক নামের কম্পিউটার ভাষাটা কতটা অশ্লীলতা সেটা খেয়াল না-করে। বেসিকের কথা তো আগেই বলেছি আপনাদের। এখানেই শেষ হয়নি ট্যানেনবম-এর আপত্তি, আবার এল তিরিশে জানুয়ারি। সকন্টেক্সট এই গোটা তর্কাতর্কিটাই আপনারা পাবেন ওরিলির ভয়েসেস ফ্রম দি ওপন সোর্স রিভলিউশনের অ্যাপেনডিক্স এ-তে, http://www.oreilly.com/catalog/opensources/book/appa.html। অন্য অনেকের মতামতের চেয়ে, স্বাভাবিকভাবেই মিনিক্স প্রণেতা ট্যানেনবমের মতামতের গুরুত্ব টরভাল্ডজ বা অন্যান্য লিনাক্সীদের কাছে অনেক বেশি ছিল। তাই আঘাতের প্রকোপটা ভালোই ছড়িয়েছিল। যাইহোক, ইতিহাসই প্রমাণ করেছে, ট্যানেনবম নয়, ঠিক ছিলেন টরভাল্ডজ। বিরানব্বইয়েই চালু হল লিনাক্সের প্রথম বাণিজ্যিক সংস্করণ — সুজে লিনাক্স। আর এই একই বছরে এক্স-উইনডোজ মানে গুই বা গ্রাফিকাল ইউজার ইন্টারফেস ঢুকল গ্নু-লিনাক্স অপারেটিং সিস্টেমে। এর আগে অব্দি ছিল শুধুই কমান্ড মোড।

১৯৯৩।। লিনাক্সীর সংখ্যা এদিকে বেড়েই চলল। তিরানব্বয়ে এলো আরো দুটো বাণিজ্যিক সংস্করণ রেডহ্যাট আর ডেবিয়ান।

১৯৯৪।। চুরানব্বইয়ে এল আরো একটা কমার্শিয়াল ডিস্ট্রিবিউশন — ক্যালডেরা। এই বছরেই চালু হল প্রথম লিনাক্স জার্নাল।

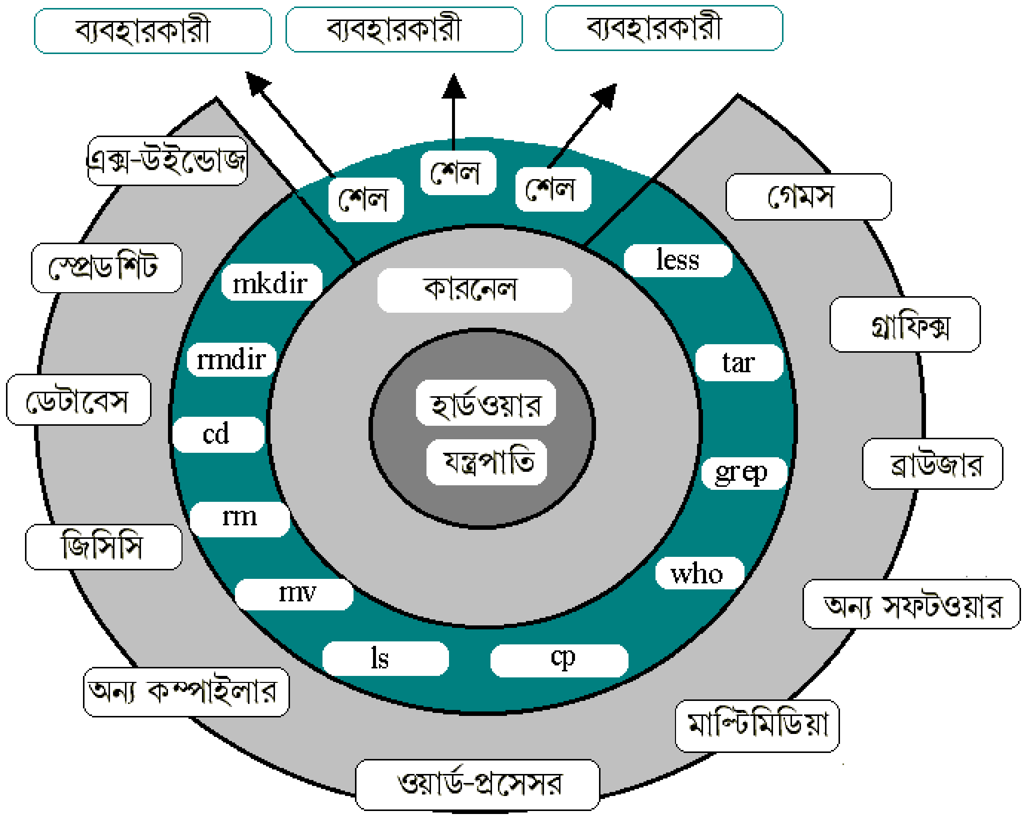
১৯৯৫।। এলো লিনাক্সের কারনেল ১.২, তার সাইজ তখন আড়াইশো কেবির উপর, যেখানে, টরভাল্ডজ-এর প্রথম ঘোষণার সময়, আকারটা ছিল মাত্র ৬৩ কেবি। কারণ, নতুন নতুন ধরণের হার্ডওয়ার, নতুন নতুন গঠনের ড্রাইভার নিয়ে আসতে হচ্ছে কারনেলের ভিতর। গ্নু-লিনাক্সের ব্যবহার তো বেড়েই চলেছে। নতুন নতুন মেশিনে, নতুন নতুন গঠনের মেশিনে।

১৯৯৬।। ফ্রি-তে বন্টনযোগ্য সফটওয়ারের কনফারেন্স হল, কনফারেন্স শুরু হল লিনাস টরভাল্ডজ-এর বক্তব্য দিয়ে, শেষ হল আরএমএসের বক্তৃতায়। এই বছরেই এটিঅ্যান্ডটি মামলা করল ক্যালিফোর্নিয়া বিশ্ববিদ্যালয়ের বিরুদ্ধে, কারণ তারা ইউনিক্সের কপিরাইট মানছে না। এই বছরের এপ্রিল মাসে লস অ্যালামস ন্যাশনাল ল্যাবরেটরি গ্নু-লিনাক্স ব্যবহার করে তৈরি করল ৬৮টা সংযুক্ত পিসির একটা সুপার কম্পিউটার, যাতে পারমানবিক শক ওয়েভের সিমুলেশন বা নকল তৈরি করা যায়। এবং চালু অন্য যে কোনো প্রযুক্তির চেয়ে এতে খরচ পড়ল অনেক অনেক কম। গ্নু-লিনাক্স ক্রমে তার নিজের জায়গা বানিয়ে নিতে শুরু করল। সিরিয়াস প্রযুক্তি, সিরিয়াস বিদ্যাচর্চায় যেমন, তেমনি একা একলা ব্যক্তিগত পিসিতেও।

টরভাল্ডজ-এর তৈরি প্রথম লিনাক্স ফ্রি কারনেলের পর প্রায় দেড় দশক চলে গেছে। এখন সেটা জ্যান্ত গ্নু-লিনাক্স দুনিয়ার প্রাত্যহিকতা। সেটা আর এইরকম লিপিবদ্ধ ক্রমপঞ্জী থেকে নয়, জ্যান্ত রকমে জ্যান্ত কাজ করে জানার। গ্নু-লিনাক্স শিখুন আর জানতে থাকুন, তৈরি করতে থাকুন সেই প্রাত্যহিকতাটা। এবার আমরা যাব সরাসরি একটা গ্নু-লিনাক্স অপারেটিং সিস্টেমের কাজ করার পদ্ধতির বিষয়ে একদম শুরুর আলোচনায়।

৪।। কারনেল আর শেল — অপারেটিং সিস্টেমের শাঁস আর খোসা

আগেও আমরা বলেছি, গোটা ইউনিক্স এবং গ্নু-লিনাক্স কাঠামোয় গোটা প্রক্রিয়াটাকে শাসন করে দুটো গুরুত্বপূর্ণ সত্তা, তারা হল কারনেল আর শেল। শেল কথা বলে সরাসরি ব্যবহারকারীর সঙ্গে। ব্যবহারকারীর প্রয়োজন মত তার পছন্দের প্রয়োগমূলক প্রোগ্রাম বা অ্যাপ্লিকেশনগুলোকে জাগিয়ে তোলে শেল। আগেই বলেছি, একটা প্রোগ্রামের একবার চলার নাম একটা প্রক্রিয়া বা প্রসেস। এদের বলা যায় ইউজার প্রসেস। আবার এই প্রোগ্রামগুলো চালাতে গেলে সিস্টেমের ভিতরে যেসব আভ্যন্তরীণ প্রক্রিয়া চালু হওয়া দরকার, সিস্টেম প্রসেস, তাদেরও চালু করে মূলত শেল। এবার এই ইউজার প্রসেস এবং সিস্টেম প্রসেসদের চলতে গেলে যে যে রসদ প্রয়োজন, সে হার্ডডিস্ক হোক, প্রিন্টার হোক, মেমরি হোক, যাই হোক — সেই প্রয়োজনের তালিকাটা তৈরি করা এবং কারনেলের কাছে পৌঁছে দেওয়াটাও শেলের কাজ। প্রয়োজনীয় এই ভৌত রসদগুলোর উপর কিন্তু শেলের কোনো সরাসরি নিয়ন্ত্রণ নেই। তাকে আর্জি জানাতে হয় কারনেলের কাছে। আর কারনেল কথা বলে সরাসরি হার্ডওয়ার বা ভৌত যন্ত্রপাতির সঙ্গে। শেলের চালু করা প্রক্রিয়াগুলোর জন্যে রসদ ব্যবহারের আর্জি মেটায় কারনেল, সেই সেই রসদের ডিভাইস ড্রাইভারের কাছে আদেশ পৌঁছে দিয়ে — দেখেছি আমরা, দুই নম্বর দিনে। কারনেলের কাছে জানানো আর্জির যা যা ফলাফল, এবং ইউজার প্রসেসদের চলার যে ফলাফল তাদের ফের ইউজারের কাছে পৌঁছে দেয় শেল। মানে মূল কাজটুকু করে ভৌত উপাদানেরা, তাদেরকে নিয়ন্ত্রণ করে কারনেল। কারনেল এবং প্রসেসগুলোর ভিতরে, তথা কারনেল এবং ইউজারের ভিতরে মধ্যস্থতা করে শেল। এখনই গোটাটা স্পষ্ট হবেনা, একটু একটু করে হতে থাকবে, এবং এটাই আবার আমাদের এই পাঠমালার শেষ মানে দশ নম্বর দিনের বিষয়, সেখানে গিয়ে আমরা অনেকটাই বুঝতে পারব। এবার শেল আর কারনেলের এই সামগ্রিক দেওয়া-নেওয়ার ছবিটা দেখুন। একদম কেন্দ্রীয় ছোট্ট বৃত্তটা হার্ডওয়ারের। তাকে ঘিরে আছে কারনেল, শাঁস। গোটা অপারেটিং সিস্টেমের শাঁসটুকু হল কারনেল। তার উপরের স্তরে রয়েছে বিভিন্ন কমান্ড আর শেল। কারনেল যদি শাঁস হয়, শেল হল অপারেটিং সিস্টেমের খোসা। যে খোসার গায়ে ব্যবহারকারীরা মানে আমরা আঁচড় কাটি, বদলাই, কাজ করি।



এখানে আমরা কয়েকটা খুব প্রাথমিক কমান্ডকে উল্লেখ করেছি। ‘mkdir’, ‘rmdir’, ‘cd’, ‘rm’, ‘mv’, ‘cp’, ‘ls’, ‘who’, ‘grep’, ‘tar’, ‘less’ — এরা খুব সাধারন প্রাত্যহিক ছোটখাটো কাজ করে। মেক-ডির বা

'mkdir' নতুন ডিরেক্টরি বানায়। রিমুভ-ডির বা 'rmdir' ডিরেক্টরি ওড়ায় — অবশ্য যদি সেটা ফাঁকা হয়। চেঞ্জ-ডির বা 'cd' এইমুহূর্তে আছি, কাজ করছি যে ডিরেক্টরিতে সেটা বদলায়, মানে একটা ডিরেক্টরি থেকে অন্য ডিরেক্টরিতে যায়। রিমুভ বা 'rm' ফাইল ওড়ায়, বা, অন্যকিছু বাড়তি শর্ত সঙ্গে দিলে ডিরেক্টরিও ওড়ায়। মুভ বা 'mv' ফাইল বা ডাইরেক্টরি এক জায়গা থেকে অন্য জায়গায় নিয়ে যায় বা তাদের নাম বদলায়। লিস্ট বা 'ls' কোনো একটা ডিরেক্টরিতে কী কী ফাইল বা সাবডিরেক্টরি আছে তার তালিকা বা লিস্টটা দেখায়। কপি বা 'cp', ফাইল কপি করে। হু বা 'who' দেখায় কে কে সেইমুহূর্তে সিস্টেমে ঢুকে রয়েছে, কাজ করছে — মানে, টেকনিকালি বললে, লগ-ইন করে রয়েছে। কোনো একটা বিশেষ শব্দ অক্ষর বা তাদের কোনো একটা সমষ্টি কোনো একটা বা একাধিক ফাইলে আছে কিনা সেটা খুঁজে দেয় গ্রেপ বা 'grep'। টার বা 'tar' হল এক বা একাধিক ফাইল বা ডিরেক্টরিকে একটা সিন্দুক ফাইলে পরিণত করার কমান্ড। আর লেস বা 'less' হল একটা পেজার — কোনো একটা ফাইলকে পাতা বাই পাতা পরপর দেখিয়ে যায় লেস। ব্যস্ত হবেন না, আলাদা করে কোনো কমান্ডকে মনে রাখার চেষ্টা করবেন না, প্লিজ। নিজের গ্নু-লিনাক্স বাক্সের সঙ্গে সময় কাটান, গ্নু-লিনাক্সে নিজেকে সংক্রমিত হতে দিন, দেখবেন, আপনা থেকেই মনে থাকবে। বরং মনে রাখার চেষ্টা করলেই টেনশন হবে, তাতে আরো ভুলবেন। আর এত কমান্ড রয়েছে গ্নু-লিনাক্সে যে কাজের মাধ্যমে তাদের সঙ্গে অভ্যস্ত হওয়া ছাড়া মনে রাখার কোনো উপায় নেই। কাজ করতে থাকুন নিজের গ্নু-লিনাক্স সিস্টেমে।

এবার দেখুন, এই কমান্ডগুলো বা ইউজার হিশেবে চালানো যে কোনো প্রোগ্রাম আমরা যখনি চালাচ্ছি, আমরা সরাসরি কথা বলছি শেলের সঙ্গে, ব্যবহারকারী বা ইউজার হিশেবে। তাদের আর কারনেলের মধ্যে যোগাযোগ করাচ্ছে শেল। কারনেল তাদের ভৌত উপাদানের প্রয়োজন মেটাচ্ছে। 'cp' ইত্যাদি খুব সহজ কমান্ডও যদি ধরেন, তাকেও তো একটা ফাইল ডিস্ক থেকে পড়তে হচ্ছে, এবং তার একটা প্রতিরূপ ডিস্কে লিখতে হচ্ছে। তার মানে ন্যূনতম হলেও ডিস্ক আর মেমরি এই দুটো ভৌত রসদ তাকে কাজে লাগাতেই হবে। রসদের এই প্রয়োজনটা মেটায় কারনেল। সবশেষে আমাদের দেওয়া কাজের ফলাফল আবার আমাদের কাছে পৌঁছে দেয় শেল। ধরুন, আমরা টাইপ করার সময় ভুল নাম দিয়েছি। এবার, শেল তো সেটা জানেনা, সে কপি করার প্রোগ্রামকে জাগিয়ে তুলল। কপি করার প্রসেস ডিস্ক পড়তে চাইল, কারনেল ডিস্ক ড্রাইভারকে জাগাল, ডিস্ক পড়ে দেখা গেল ওই নামে ওখানে কোনো ফাইলই নেই। তখন শেল আবার আমাদের কাছে ফেরত এল, এবং আমাদের জানাল, কপি করা যায়নি, কারণ, যে নামের ফাইল কপি করতে চাইছি, সেই নামে ডিস্কে কোনো ফাইলই নেই। আর যদি ওই নামে কোনো ফাইল থাকে, তাহলে কিছুই বলল না। গ্নু-লিনাক্সে কাজ করার সময় এটা খেয়াল রাখবেন। শেল কিন্তু অত্যন্ত গম্ভীর রিজার্ভ বাকসংযমী। কোনো কাজের কমান্ড দেওয়ার পর কিছুই না জানানো মানে কাজটা নির্বিঘ্নে হয়ে গেছে। এই গোটাটাকেই দেখানো আছে ছবিতে।

একটু বিশদ করে বোঝার চেষ্টা করা যাক এবার, ইউজার এবং শেলের মধ্যের সম্পর্কটাকে। তার জন্যে প্রথমে আমরা কমান্ড দেওয়ার প্রক্রিয়াটাকে বুঝি, আসুন। ধরুন আপনি প্রবন্ধ লিখছেন। প্রথমে আপনি নাম দিয়েছিলেন 'essay.text', তখনো ভেবেছিলেন, একটাই প্রবন্ধ হবে। এবার দেখলেন এই নিয়ে আপনার আরো প্রবন্ধ লেখার আছে। কী নাম দেবেন, নতুন করে ভাবলেন। এবার ভাবলেন নাম দেবেন এক দুই করে, 'essay.1.text', 'essay.2.text' ... ইত্যাদি। এবার, পুরোনো প্রথম প্রবন্ধটার নাম আপনাকে বদলাতে হবে। এর জন্যে কমান্ড বা আদেশ দিলেন, '`mv essay.text essay.1.text`'। কিন্তু আপনার লেখা তো থেমে নেই, এবার মনে হল এই প্রবন্ধের সিরিজে আর কয়েকটা লিখলে তো একটা বই হতে পারে, তাই, এই সমস্ত প্রবন্ধ ফাইলগুলো আলাদা একটা ডিরেক্টরিতে আনবেন ঠিক করলেন, যার নাম দেবেন 'book.one'।

আপনি কমান্ড দিলেন, '`mkdir book.one`'। অ্যাদ্দিনের সব প্রবন্ধগুলোকে এবার ওই ডিরেক্টরিতে পাঠাতে হবে, কমান্ড দিলেন '`mv essay*text book.one`'। এই *-এর চক্করটা একটা রেগুলার এক্সপ্রেশন, মানে যে বঁড়শিতে একসঙ্গে অনেক মাছ তোলা যায় এমন একটা পাইকিরি হুক, দশ নম্বর দিনে খুব ভালো করে আসব আমরা এটায়। *-এর আগের আর পিছের দুটো অংশের গঠনটা আপনি দিয়েছেন, মধ্যে *-এর জায়গাতে যাই থাকনা কেন, সেই নামের ফাইল পাঠিয়ে দেওয়া হবে ওই 'book.one' ডিরেক্টরিতে। এক জায়গায় আপনার সব প্রবন্ধের ফাইলগুলো পাওয়া যাবে।

এবার ধরুন, বইটা শেষ হয়েছে, আপনার এক বন্ধুকে পাঠাবেন বইটা, মেল করে, কিন্তু মাথায় আছে, টেলিফোন বিল, মাসের শেষের অশান্তি, অর্থনৈতিক ধ্বস ইত্যাদি। আপনি তাই গোটা ডিরেক্টরিটাকে একটা সিন্দুক বানিয়ে যারপরনাস্তি ছোট করতে চাইলেন। কমান্ড দিলেন '`tar -cjf book.one.tar.bz2 book.one`'। এতে বইয়ের গোটা 'book.one' ডিরেক্টরিটা সিন্দুক হয়ে ছোট্ট হয়ে অনেক বাইট সাইজ কমে নতুন একটা ফাইল তৈরি হল যার নাম 'book.one.tar.bz2'। এইটা ইমেল করে অনেক কম খরচেই পাঠাতে পারবেন বন্ধুকে। সে আবার পড়ার আগে উল্টো একটা কমান্ড দিয়ে সেটাকে ভেঙে নেবে। এই শেষ কমান্ডটায় আমরা আসলে দুটো কমান্ডকে একই সঙ্গে ব্যবহার করেছি, একটা হল 'tar', আগেই বলেছি, আর অন্যটা হল 'bzip2'। পরে শিখে যাবেন, গ্নু-লিনাক্স করতে করতে, না-শিখে মেলে বই পাঠাতে গেলে আপনার বাড়ি থেকে আপনাকে নির্দ্বিধায় 'কেয়ার অফ ফুটপাথ' করে দেবে। এই দুটো কমান্ডের কোনো একটাকে জানার যদি খুব কৌতূহল হয়, তাহলে আপনার গ্নু-লিনাক্স সিস্টেমে দাঁড়িয়ে কমান্ড দিন, '`man tar`' বা '`man bzip2`'। সিস্টেম আপনাকে এই দুটো কমান্ডের ম্যানুয়াল পাতাটা খুলে দেবে। এরকম যে কোনো কমান্ডেই আপনি পড়ে নিতে পারেন। শুধু পারেন না, পড়তে আপনাকে হবেই, যদি কাজ করতে চান।

এর প্রত্যেকটা কমান্ড আপনি কিবোর্ড থেকে টাইপ করে দিয়েছেন, কমান্ড প্রম্পটে। কমান্ড প্রম্পট (Command Prompt) মানে টাইপ করার জায়গা। যদি আপনি এক্স-উইনডোজ বা গুই-তে থাকেন, তাহলে একটা টার্মিনাল অ্যাপ্লিকেশন উইনডো। আর যদি এমনিতে থাকেন, তাহলে সিস্টেম বুট হওয়ার পর কালো যে জায়গাটা ফুটে ওঠে, কার্সর জ্বলতে নিভতে থাকে, আপনি কিছু টাইপ করলে সেটা যেখানে ফুটে ওঠে। মেশিনকে কোনো আদেশ বা কমান্ড দেওয়া মানে এই কমান্ড প্রম্পটে কমান্ডটা টাইপ করা, তারপর এন্টার (Enter) সুইচটা টিপে সেই আদেশটা সিস্টেমে ঢোকানো। যেই আপনি আদেশটা দিয়ে এন্টার মারেন, সঙ্গে সঙ্গে ওই জায়গাটা হাওয়া হয়ে যায়, কমান্ড প্রম্পটটা। যতক্ষণ সে গায়েব থাকছে আপনার আর কিছু করার নেই। কমান্ড প্রম্পট গায়েব মানেই সিস্টেম কাজ করছে। আপনার মেশিন কাজ করছে, জোনাকির মত আপনার মেশিনের টাওয়ারের গায়ে লাগানো হার্ড ডিস্ক কাজ করার কুচো বাল্বটা কিছু সময় ধরে তার কুতকুতে চোখ টিপটিপ করে চলেছে। তারপর আবার আপনার কমান্ড প্রম্পটটা ফেরত এল। মানে, মেশিনের কাজ করা শেষ হয়েছে, আবার আপনি নতুন আদেশ দিতে পারেন। তবে ছোটখাটো কাজ এত দ্রুত হয় যে এই ভ্যানিশ করে যাওয়া এবং ফেরত আসাটা খুব খেয়াল না-করলে বোঝাই যায়না। তার মানে দেখুন, আপনার কমান্ডটা পালন করতে গিয়ে আপনার মেশিনকে কিছু কাজ করতে হল তার ভৌত রসদদের নিয়ে, সে হার্ডডিস্ক, অন্য কোনো ডিস্ক, মেমরি, ইনপুট-আউটপুট, যাই হোক।

একটু আগের সিন্দুক বানানোর কমান্ডটা ভাবুন, 'tar -cjf book.one.tar.bz2 book.one'। আপনার আদেশে কী ছিল? 'book.one' ডিরেক্টরিতে আপনার কিছু প্রবন্ধ, 'essay.1.text', 'essay.2.text' ... ইত্যাদি নামের কিছু ফাইল। যার প্রত্যেকটা ফাইলে আপনার একটা করে প্রবন্ধ আছে। মানে আসলে আছে কিছু বাইট, সিস্টেম যাদের বর্ণমালা এবং যতিচিহ্ন করে আমাদের সামনে হাজির করেছে। এবার, 'book.one' নামের গোটা ডিরেক্টরিটাকে তার যাবতীয় ফাইল সমেত 'tar' নামের প্রোগ্রাম দিয়ে প্রথমে একটা সিন্দুক ফাইল বানানো হচ্ছে। ডিরেক্টরিময় সমস্ত ফাইল তখন সেই একটামাত্র ফাইলে পরিণত হল। এবার সেই সিন্দুকটাকে আবার ছোট করে একটা কোঁকড়ানো সিন্দুক বা কমপ্রেসড টার ফাইল বানানো হচ্ছে। 'book.one.tar.bz2' নামের সেই কোঁকড়ানো সিন্দুকটাকে ডিস্কে লিখে ফেলার পর তবে আপনার কমান্ডের মোট কাজ শেষ করতে পারবে আপনার সিস্টেম।

এই গোটাটা জুড়ে আপনার সিস্টেম কী কী করছে? প্রথমে, গোটা ডিরেক্টরিটা সবগুলো ফাইল সমেত পড়তে হচ্ছে। পড়া মানে কী? ওই সমস্ত ফাইলের সমস্ত বাইটের প্রতিরূপ তুলে মেমরিতে রাখা। এবার, 'tar' আর 'bzip2' প্রোগ্রামদুটো সিস্টেম কাজে লাগাতে গেলে আগে গোটা প্রোগ্রামদুটোর মধ্যে লেখা আদেশের বাইটগুলোকে মেমরিতে তুলে নিতে হবে, তারপর চালাতে হবে। তখন দুটো আপাতমৃত প্রোগ্রাম ফাইল জ্যান্ত হবে, তাদের আমরা ডাকব দুটো প্রসেস বলে — মনে পড়ছে? তারপর 'tar' আর 'bzip2' নামক প্রোগ্রামের আদেশতালিকা মোতাবেক ইতিমধ্যেই মেমরিতে তুলে নেওয়া 'book.one' ডিরেক্টরির যাবতীয় ফাইলের বাইট সমগ্রকে তুমুল চটকানো। চটকাতে চটকাতে যখন গোটাটা যখন ডিস্কে বেলে ফেলার মত জায়গায় আসবে তখন চটকানো শেষ। মেমরিবাসী সেই চটকে ফেলা বাইটসমগ্রকে এবার ডিস্কে পাঠানো, 'book.one.tar.bz2' ফাইলে লিখে ফেলার জন্যে, তবে কাজ

শেষ মেশিনের। আসলে কিন্তু জনান্তিকে, পর্দার পিছনে আরো বহু বহু কিছু ঘটছে, কিছুটা যার আন্দাজ আমরা আগেই দিয়েছি সিস্টেমের ডিমন বা যখদের কথায়, কিম্বা সিস্টেম প্রসেসের প্রসঙ্গে। এখানে আমরা সেটাকে ছোট করে নিলাম, জঙ্গল দেখতে গিয়ে যাতে গাছ হাওয়া না-হয়ে যায়। অর্থাৎ, আপনার মেশিনের ভৌত রসদগুলোর ভিতর অন্তত হার্ডডিস্ক আর মেমরি তো ব্যবহার করতেই হবে, আপনার আদেশ পালন করতে গেলে। কিন্তু।

কিন্তু আপনি কখনোই সরাসরি কারনেলের সঙ্গে, মানে সব রসদের ভান্ডারির সঙ্গে কথা বলেননি, বলেছেন শেলের সঙ্গে। মিডলম্যান হয়ে শেলের এই কাজটা খুব স্পষ্ট বোঝা যায় আপনার ওই অনেক মাছ একত্রে গাঁথার বড়শি মানে রেগুলার এক্সপ্রেশনটা দেখলে। আপনি তো বঁড়শিটা লাগিয়েই খালাস হয়েছিলেন, তার সঙ্গে মিলিয়ে মিলিয়ে কোন কোন মাছ ওই বঁড়শিতে গাঁথা হবে, আর কোনগুলো পরের বারের জন্যে রেখে দেওয়া হবে সেটা বলে দিল কে? আগেই তো বলেছি আমরা কারনেল হার্ডওয়ারের উপরে ছেয়ে দেওয়া একটা প্রলেপের মত, ওইসব তার কাজ নয়, তার কাজ ভৌত উপাদানদের নিয়ন্ত্রণ। এই কাজটাই করে শেল। এবং গ্নু-লিনাক্সে বেশির ভাগ ক্ষেত্রেই এই শেলটার নাম ব্যাশ (BASH — **B**ourne-**A**gain-**SH**ell) — কিন্তু এছাড়াও কর্ন শেল, জেড শেল, ইত্যাদি অনেকরকম হতে পারে।

আমরা বলেছিলাম, এই কমান্ডগুলো বা তাদের কোনো সমবায়, যেমন একটা খুব সরল সমবায় এই মাত্র আমরা দেখলাম, 'tar' আর 'bzip2', কাজ করে কারনেলের উপর দাঁড়িয়ে। আর শেল তৈরি করে দেয় আমাদের আর কমান্ডের মধ্যে কথা বলার জায়গাটা। আমরা এককথায় সুইপ করে বেরিয়ে গেলাম — এই কমান্ডগুলো বা তাদের কোনো সমবায় — এখানে পরে আমাদের বিপুল ভাবে ফিরে আসতে হবে, বারবার, এই সমবায় ধারণাটা হল ইউনিক্স লিনাক্স-এর সবচেয়ে জোরের জায়গার একটা, তার দার্শনিক ভিত্তিগুলোর একটা। আমরা যখন ইউনিক্সের ক্রমপঞ্জীতে প্রথম দিকের ইউনিক্সে পাইপ না-থাকার কথা বলেছিলাম, সেই পাইপ হল যাদের দিয়ে এই সমবায়টা করা হয় তাদের মধ্যে একটা জোরালো অস্ত্র।

আপনারা যারা উইন্ডোজ পরিবেশে কাজ করতে অভ্যস্ত, সেখানে দেখবেন, কোনো সফটওয়ার ভালো হচ্ছে উন্নত হচ্ছে মানেই আরো গাবদা হচ্ছে, বিএমআই যাচ্ছে পঁচিশ ছাড়িয়ে, কোমর যাচ্ছে পঁয়ত্রিশ ছাড়িয়ে, ভুঁড়ি বেড়ে যাচ্ছে বেল্ট ছাড়িয়ে — মানে আরো আরো পোকা বা বাগ গজাচ্ছে। এর ঠিক উল্টো দর্শন থেকে শুরু হয়েছিল ইউনিক্স-এর যাত্রা, গ্নু-লিনাক্স তাকে আরো বাড়িয়ে তুলেছিল। সেটা হল, যে কোনো একটা কমান্ড খুব ছোট একটা পরিসরে কাজ করবে, কিন্তু সেই কাজটুকু করবে অপ্রশ্নেয় দক্ষতায়। আর আরো আরো জটিল কাজ, বড় কাজ, ঘুরপথে কাজ, বেঁকা কাজ মানে জাস্ট ওই কুচো কুচো কিন্তু ধারালো কমান্ডগুলোকে নাও, তাদের একটার সঙ্গে অন্যটাকে জোড়ো, আর একটাকে, আর একটাকে, প্রয়োজন পড়লে জুড়েই চলো। এবার যেটা পাওয়া যাবে সেটা বড় কাজই করবে, কিন্তু একা একটা কোনো কমান্ড দিয়ে না, কমান্ডের সমবায় দিয়ে। এই মুহূর্তে আপনাদের কারোর কারোর মনে হতে পারে, এদুটোর মধ্যে এমন বিরাট পার্থক্যটা কোথায়, কিন্তু সেই উত্তরটার জন্যে আমাদের এগোতে হবে আরো অনেকটা।

ধরুন, একটা প্রাথমিক আন্দাজ দিয়ে নেওয়া যাক, এর গোটাটা আপনি এই মুহূর্তে নাও বুঝতে পারেন, চিন্তিত হবেন না, ধারণাটা বোঝার চেষ্টা করুন। পরে এর সবগুলো উপাদানই আমরা অনেক ভালো করে আনব। একটু আগে আমরা যে কমান্ডটা দিয়ে আপনার আস্ত বইটাকে একটা কোঁকড়ানো সিন্দুক বানিয়েছিলাম, সেই কমান্ডটাই ভাবুন,

```
tar -cjf book.one.tar.bz2 book.one
```

এর মধ্যে '`j`' অংশটুকু আসলে বিজিপটু-র কাজ করছে। এই কমান্ডটাকে আমরা এইরকম ভেঙেও লিখতে পারতাম,

```
tar -cf - book.one | bzip2 > book.one.tar.bz2
```

এর মধ্যে দেখুন তিনটে আলাদা অংশ আছে, '`tar -cf - book.one`' এবং '`bzip2`' এবং '`book.one.tar.bz2`'। এখানে আমরা এই সমবায় টেকনোলজির দুটো জায়গা ব্যবহার করেছি, পাইপ এবং রিডাইরেকশন, পথ-দেখানো এবং চালান করা। প্রথমটার চিহ্ন '|' আর দ্বিতীয়টার চিহ্ন '>'। প্রথম চিহ্নটার বাঁ পাশের অংশটা book.one নামক ডাইরেক্টরিকে '`tar -cf - `' করে সিন্দুক বানাতে বলছে। এই অব্দি করে আমরা যদি ছেড়ে দিতাম তাহলে স্ক্রিনময় হুড়হুড় করে নেমে যেতে থাকত ওই রকম সব চিহ্ন, শূন্য নম্বর দিনে ভিশুয়াল চ্যাংড়ামিতে আমরা যা দেখিয়েছিলাম একটা পিডিএফ ফাইলকে টেক্সট এডিটর দিয়ে খুলে। কিন্তু স্ক্রিনে

ওই অপদেবভাষা দেখে আমাদের আর কী পুণ্য হবে? তাই বেচারাদের আমরা পথ দেখালাম, পাইপ করলাম, পাঠিয়ে দিলাম দ্বিতীয় অংশের কাছে, যেখানে সেই সিন্দুকটাকে কুঁকড়ে ফেলবে। কোনো আসুরিক শক্তি দিয়ে নয়, তার কায়দা আছে অনেক, এবং বেশ মজার কায়দা গুলো। ধরুন 'কালো ভালো আলো মালো' এদের আপনি লিখলেন 'কাভাআমা৪লো'। এটা একটা জঘন্য এবং তড়িঘড়ি, কুইক অ্যান্ড ডার্টি, উদাহরণ হল। কিন্তু আন্দাজটা পেলেন বোধহয়? যাইহোক, কোনো অজঘন্য এবং শ্লথ পদ্ধতিতে সিন্দুকটাকে কোঁকড়ানো হল। এই গোটাটাই ঘটল মেমরিতে। সরাসরি কাঁচা মেমরিতে বা ভারচুয়াল মেমরির অস্থায়ী ফাইলে। তারপর মেমরিবাসী বাইটগুলোকে চালান করা হল, রিডাইরেক্ট করা হল হার্ডডিস্কের নির্দিষ্ট কিছু সেক্টরে, যাদের জন্যে নাম সাব্যস্ত করে দেওয়া হয়েছে আগেই, 'book.one.tar.bz2'। এই গোটা প্রক্রিয়াটাকে আমরা শর্টকাটে আর একটা রকমে লিখেছিলাম

```
tar -cjf
```

যাইহোক, এখুনি এর গোটাটা বোঝা সম্ভব না, কিন্তু আন্দাজটা তৈরি করে নিন। যাতে কারনেল এবং শেলের গোটা কাজের পদ্ধতিটাকে বুঝতে পারেন।

আর ইউনিক্স জগতের সেই বিখ্যাততম পাইপ প্রদর্শনীটা তো আছেই, যেখানে আপনি সিস্টেমে আপনি সহ মোট কজন এই মুহূর্তে কাজ করছেন সেটা বোঝার জন্যে দিচ্ছেন

```
who | wc -l
```

এর মধ্যে 'who' অংশটা প্রথমে তৈরি করছে তার কাজের ফলাফল, মানে যারা সিস্টেমে কাজ করছে বা লগ-ইন করে রয়েছে তাদের তালিকা। 'who' তালিকাটা তৈরি করে লাইন পিছু একজন করে ইউজার বা ব্যবহারকারীকে রেখে, সেটাই 'who' প্রোগ্রামের নিয়ম। এবার 'who' প্রোগ্রাম তার কাজের ফলাফলটাকে পাঠিয়ে দিচ্ছে দ্বিতীয় অংশের কাছে, যেখানে 'wc -l' অংশটা তার লাইন গুনছে। যে কোনো টেক্সট ফাইলের উপর 'wc' চালালে, 'wc' ফাইলটার মোট লাইনের সংখ্যা, শব্দের সংখ্যা, এবং চিহ্নের সংখ্যা পরপর ফুটিয়ে তোলে। আর 'wc -l' দেখায় শুধু লাইনের সংখ্যা। এবার, 'who' প্রোগ্রামের ফলাফলটার লাইন গোনা মানে, আপনি যখন দ্বিতীয় অংশটার কাজের ফলাফল পাচ্ছেন, সেটা একটা সংখ্যা, ব্যবহারকারীদের মোট সংখ্যা। শুধু 'who' করলে পাওয়া তালিকা থেকে যা আপনি নিজেই গুনে নিতে পারতেন। এই গোটা কাজটাই ভাবুন। করে দিচ্ছে শেল, আপনার হাত থেকে গোটা কমান্ডটা সে পড়ে নিচ্ছে, তারপর তার মধ্যের বিভিন্ন অংশকে আলাদা আলাদা ভাবে ঘটিয়ে নিচ্ছে। ঘটাতে গিয়ে তাকে অনেক কিছু করতে হচ্ছে, কারনেলের কাছে আবেদন ইত্যাদি, কারণ, কাজ করা মানেই তো রসদ ব্যবহার করা, তারপর মোট কাজের ফলাফলটাকে ফের প্রোগ্রামগুলোর কাছ থেকে ফিরিয়ে আনা এবং স্ক্রিনে ফুটিয়ে তোলা। এই গোটা পদ্ধতিটা আজ আমরা প্রাথমিক রকমে বুঝব। আট নয় দশ নম্বর দিনে আরো অনেক কিছু জানার আছে।

কারনেল এবং তার চারদিকে কাজ করতে থাকা শেল আর কমান্ডগুলোকে মিলিয়ে তৈরি হয় অপারেটিং সিস্টেম। তার উপর দাঁড়িয়ে আছে প্রয়োগ সফটওয়ারগুলো, যেমন, গুই বা এক্স-উইনডোজ — সেটা নিজেও একটা অ্যাপ্লিকেশন, আবার তার উপর দাঁড়িয়ে কাজ করে অনেক অ্যাপ্লিকেশন। অন্য প্রয়োগ সফটওয়ারগুলোর ভিতর খুব বেশি ব্যবহার হয় স্প্রেড-শিট বা হিশেবনিকেশের সফটওয়ার, ডেটাবেস বা কোনো রাশি বা তথ্যের সমাহারকে নাড়াচাড়া এবং ব্যবহার করার সফটওয়ার, জিসিসি মানে আমরা জানি — সি বা সিপ্লাসপ্লাস জাতীয় হাইলেভেল ভাষায় লেখা প্রোগ্রামকে লোলেভেল মেশিনোপযোগী করে তোলার কম্পাইলার, ওয়ার্ড-প্রসেসর — যা দিয়ে আমরা লিখি, এই লেখাটাও লিখছি, গ্রাফিক্স — যা দিয়ে ছবি আঁকি, একটু আগে শেল আর কারনেলের ছবিটা আঁকলাম, ব্রাউজার মানে যা দিয়ে নেটে আমরা একটা ওয়েবসাইট থেকে আর একটা ওয়েবসাইট টুঁড়ে বেড়াই — এত সব যে লাগাতার পন্ডিতি মেরে যাচ্ছি তার খনি তো ওই নেটেই, বা, মাল্টিমিডিয়া মানে গান-সিনেমা-অ্যানিমেশন ইত্যাদি।

আগের ছবিতে দেখুন শেল কারনেল আর হার্ডওয়ারে বিন্যস্ত কাজের বৃত্তাকার ছক, এবং এর সঙ্গে একবার মিলিয়ে নিন এক নম্বর দিনের সেই টেবিলটা, যেখানে আমরা অপারেটিং সিস্টেমের তিনটে স্তরে স্তরবিন্যাসটা দেখিয়েছি — কারনেল আছে হার্ডওয়ারকে প্রলেপের মত ঘিরে, অপারেটিং সিস্টেমের কেন্দ্রীয় জায়গায়। এই কারনেল হল, গ্নু-লিনাক্সের ক্ষেত্রে, সি-তে লেখা কিছু কোডের সমষ্টি। যারা সরাসরি হার্ডওয়ারের সঙ্গে কথোপকথন চালায়। যার পাগলা কুকুরে কামড়ানো রকমের দামী ভার্শন বেচে আসছিল এটিঅ্যান্ডটি, আর যার প্রথম ফ্রি সংস্করণটা নেটে

তুলেছিল লিনাস টরভাল্ডজ, মাত্র তেষট্টি কেবির। এই ফ্রি মানে শুধু দামের ফ্রি নয়। সেটা প্রথম অর্থ, যে কেউ যে কেউকে এই সফটওয়ার দিয়ে দিতে পারে, কোনো আইন ভাঙা হয়না তাতে। আর দ্বিতীয় অর্থটা অনেক বেশি ব্যাপক। যে কেউ, যে বুঝতে চায় কারনেলকে, সিস্টেমকে, সে এবার বুঝতে পারবে, কোড গুলো পড়ে পড়ে, কী ভাবে সেটা কাজ করছে, কী ভাবে আদেশমালার মধ্যে দিয়ে প্রোগ্রামগুলো চলছে — কী করে কাজ করে একটা কম্পিউটার সিস্টেম। বুঝতে পারবে, ধরতে পারবে, নতুন ধরনের সিস্টেমে নতুন ধরনের কাজের স্বপ্ন দেখতে পারবে। নতুন চিন্তাকে নতুন ধরনের প্রকৌশলকে নিয়ে আসতে পারবে, যদি তার সেই আনার যোগ্যতা থাকে। মানে, জানার মুক্তি, ফ্রিডম অফ নলেজ। গেটস-এর ভাড়াটে কম্পিউটার বিজ্ঞানীরা যে অব্দি গিয়ে পৌঁছতে পেরেছে সেটাই মানুষের চিন্তার সীমা বানিয়ে তোলার স্পর্ধা বিল গেটস করতে পেরেছে, কারণ, বাজার ওভাবেই কাজ করে। সেই সীমার বাইরে মানবসভ্যতার আবহমান জানার স্বাধীনতার অর্থে ফ্রি। আমরা তিন নম্বর দিনে যে কম্পিউটার যন্ত্র এবং চিন্তার যে ধারাবাহিকতাটাকে এনেছি, তাকে যে কোনো একটা জায়গায় ভাঙুন, দেখুন পরের টুকু আর আসছে না। এতদিন অব্দি সেটা এসে থেকেছে, এভাবেই, রেডমন্ডের র‍্যাপ্টর এবং তার স্বজাতীয়দের আগে অব্দি। এখন আমাদের চিন্তার সীমা হয়ে দাঁড়াচ্ছে গেটসের গেট দিয়ে যতটুকু আকাশ দেখা যায়। তার বাইরে যাওয়ার ফ্রিডমটা হল ফ্রি-এর দ্বিতীয় অর্থটা। ফ্রি বিয়ারের ফ্রিডমটাও উমদা জিনিষ, কিন্তু জানার ফ্রিডমটা বেহাত হয়ে যাওয়াটা আরো ডেঞ্জারাস। হঠাৎ একদিন দেখা যায়, হোজে মার্তির সেই কবিতার মত, পিট সিগারের গানের মত, কবে যেন আকাশ হাওয়া জঙ্গলটাও বেহাত হয়ে গেছে।

সিস্টেম যখন বুট করে, মানে মৃত একটা টিভি স্ক্রিনের মত মনিটর আর একটা বড় লম্বা টিনের কৌটোর ক্যাবিনেটে ঘেরা টাওয়ারের নিষ্প্রাণতা থেকে জ্যান্ত কাজমান একটা অস্তিত্ব হয়ে ওঠে, প্রাণ পায়, সেই সময়ে মেমরিতে কারনেলের এই সি-কোডগুলো তুলে নেওয়া হয় একের পর এক। পরে বুট-পদ্ধতির আলোচনায় আমরা আরো ভালো করে বুঝব। সিস্টেমের মোট রসদের ভাণ্ডারটাকে ম্যানেজ করে কারনেল, বিভিন্ন ব্যবহারকারীর চালানো বিভিন্ন প্রোগ্রাম কে কখন কার পরে প্রসেসরকে কাজে লাগাবে — সেটা ঠিক করে, এবং হার্ডওয়ারের যেসব বিটকেল খুঁটিনাটি জেনে কাজ করতে হলে আমাদের বরং কিবোর্ড না-টিপে ডুগডুগি বাজাতে হত সেগুলোকে ত্রুটিহীন রকমে মাথায় রেখে নিঃশব্দে আমাদের কাজ করার ব্যবস্থা করে চলে। বিভিন্ন ধরনের মাল্টিপ্লেক্সিং নিয়ে এক আর দুই নম্বর দিনে আমাদের আলোচনাটা মনে করুন। বিভিন্ন প্রায়োগিক সফটওয়ার বা অ্যাপ্লিকেশনগুলো তাদের প্রয়োজন মোতাবেক বিভিন্ন রসদের জন্যে আবেদন জানায় — সেটা এই কার্নেলের কাছেই — তারই নাম সিস্টেম কল, আমরা আগেই বলেছি। কার্নেল হল বিভিন্ন প্রোগ্রামের মেশিনে ঢোকার প্রবেশপথ।

৫।। ভৌত উপাদানকে ভুলবেন না

উইন্ডোজ থেকে গ্নু-লিনাক্সে কম্পিউটারটা বদলায় না, তার অঙ্গপ্রত্যঙ্গ বদলায় না, বদলায় তার অপারেটিং সিস্টেম, বদলায় ওই অঙ্গপ্রত্যঙ্গ গুলোকে মানে হার্ডওয়ার ডিভাইসগুলোকে অপারেটিং সিস্টেম কী ভাবে নাড়াচাড়া করে তার ব্যাকরণটা। তার সঙ্গে সঙ্গে বদলায় নামকরণটাও, দেখার পদ্ধতিটাও। যার একটু আধটু আমরা আগেই বলেছি। ভালো করে জানব ছয় আর সাতে গিয়ে। এবার, দেখে নিন, এই অ্যানাটমিটা আপনার মনে আছে কিনা। আমরা শূন্য নম্বর দিনে এর একদম প্রাথমিক পরিচয়টা করেছিলাম, মেশিনের কাজের ছক অনুযায়ী, যা মূলত ভন নয়মান আর্কিটেকচার প্রাথমিকতম ছকটা। তারপর এক আর দুই নম্বর দিনেও এসেছে, সিস্টেম কল এবং ইন্টেরাপ্ট আলোচনার সূত্রে। পিছিয়ে গিয়ে একবার দেখে আসুন। আমরা বুঝতে চাইছি গ্নু-লিনাক্স একটা কাঠামোর কাজের পদ্ধতি, গ্নু-লিনাক্স একটা ওএস ঠিক কী ভাবে কাজ করে সেটা বুঝতে চাই আমরা। সেটা প্রধানত একটা সফটওয়ার জগতের যাত্রা। কিন্তু বারংবার সেখানে দরকার পড়ে ভৌত বাস্তবতাটাকে মাথায় রাখার।

আমরা যে তালিকাটার সঙ্গে পরিচিত হয়েছি, তার মধ্যে প্রধানতমটা অবশ্যই মাইক্রোপ্রসেসর চিপ। এর পরেই কাজের গুরুত্বে দ্বিতীয় জায়গায় আসে মেমরি। এখানে মেমরি বলতে র‍্যাম চিপ এবং রম চিপ দুই-ই বোঝাচ্ছি আমরা। প্রসেসর চিপ আর মেমরি চিপ মিলে তৈরি হয় কম্পিউটারের হৃদপিণ্ড মানে, কম্পিউট করার বা তথ্য চটকানোর মূল কাজটা যে করে। নানা ধরনের অ্যাডাপ্টার কার্ড, বৈদ্যুতিক সংযোগ ইত্যাদি সহ এই চিপগুলো লাগানো থাকে মেশিনের মাতৃক্রোড়ে, মানে মাদারবোর্ডে। তথ্য এবং বিদ্যুতের সংযোগে এদের সঙ্গে যুক্ত থাকে

হার্ডডিস্ক এবং ফ্লপিডিস্ক এবং কম্প্যাক্টডিস্ক বা সিডি গোছের স্মৃতি তুলে রাখার ভাঁড়ার। প্রয়োজন মত সেখান থেকে লিখে রাখা তথ্য পড়ে ফেলাও যায়। চটকে ফেলা তথ্য মেশিন থেকে বার করার জন্যে আছে মনিটর বা কনসোল বা স্ক্রিন এবং প্রিন্টার এবং সাউন্ডবক্স গোছের আউটপুট ডিভাইস। ওই তথ্য-ভাঁড়ার দিয়েও এই কাজটা করা যায়। আর মেশিনে তথ্য ঢোকানোর জন্যে থাকে কিবোর্ড এবং মাউস জাতীয় ইনপুট ডিভাইস। বা কোনো তথ্য-ভাঁড়ার যা সেই মূহূর্তে ইনপুট ডিভাইস হিশেবে ব্যবহার হচ্ছে।

এর বাইরেও আরো বহু কিছু থাকে মেশিনে। যেমন এই ডিভাইসগুলোর অনেকের জন্যেই লাগে কন্ট্রোলার কার্ড — যার মাধ্যমে এই ডিভাইসকে ব্যবহার করে অপারেটিং সিস্টেম। এই কন্ট্রোলার কার্ডদের লাগানো থাকে মাদারবোর্ডে। কিবোর্ড জাতীয় ডিভাইসের আলাদা করে কন্ট্রোলার কার্ড লাগেনা, কারণ সেই কার্ডটা থাকে কিবোর্ডের ভিতরেই। এগুলোর বাইরে আরো বহু রকমের ডিভাইস হয়। যেমন রেইড (RAID — **R**andom-**A**rray-of-**I**nexpensive-**D**isks) — পরপর অনেকগুলো হার্ডডিস্কে একাধিক কপিতে লেখা থাকে ফাইল ফোল্ডার ফাইলসিস্টেম। এটা করা হয় তথ্যের বাড়তি নিরাপত্তার জন্যে, একটা সেট নষ্ট হয়ে গেলেও তথ্যটা যাতে হারিয়ে না যায়। এই রেইড অ্যাক্রোনিমটায় অন্য শব্দও দেখেছি। ইনএক্সপেনসিভ-এর জায়গায় ইনফর্মেশন, র‍্যান্ডম-এর জায়গায় রিডানড্যান্ট ইত্যাদি। হয় স্কাসি (SCSI — **S**mall-**C**omputer-**S**ystem-**I**nterface) — শুধু হার্ডডিস্ক ড্রাইভ নয়, অনেকসংখ্যক অনেকরকম ড্রাইভ একসাথে ব্যবহার করার খুব উন্নত একটা প্রযুক্তি। ইত্যাদি ইত্যাদি। এর মধ্যে কিছু ডিভাইসের খুঁটিনাটি নিয়ে আমরা পরেও আলোচনা করব। বিশেষত সাত নম্বর দিনে ডিভাইস ফাইলের আলোচনায়।

আগেই বলেছি, কম্পিউটারের ভিতরে এই প্রতিটা অঙ্গপ্রত্যঙ্গই একে অন্যের সাথে তথ্যপথ বা বাস দিয়ে যুক্ত। ভৌত আকারের দিক দিয়ে বলতে গেলে, ধরুন ওই সার্কিটবোর্ডটা জুড়েই বাসের বসবাস, যে বোর্ডে আমরা গুঁজে দিয়েছি কন্ট্রোলার কার্ডগুলো। ভিডিও কার্ড — যা আমার স্ক্রিনে লেখা বা ছবি ফুটে ওঠার প্রক্রিয়াটা চালায়। সাউন্ড কার্ড — যা আওয়াজের শব্দসংকেত বা অডিও সিগনাল এবং বৈদ্যুতিক সিগনাল এই দুটোকে একটা থেকে অন্যটায় বদলায়। ডিস্ক কন্ট্রোলার মানে যা ডিস্কগুলোকে নিয়ন্ত্রণ করে। বাসই সংযুক্ত রাখে একহাতে প্রসেসর চিপ আর মেমরি, এবং অন্যহাতে ডিস্ক-ড্রাইভ, ডিভাইস, কার্ড — এই সবকিছুকে। চিপ, স্ক্রিন, ডিস্ক — সবকিছুর সঙ্গে অন্য সবকিছুর তথ্যপথটাই হল বাস। এই বাস কী ভাবে যুক্ত রাখে একদিকে সিপিইউ আর মেমরি আর অন্যদিকে পেরিফেরাল বা ভৌত উপাদানগুলোকে তার কিছু আলোচনা আমরা ইতিমধ্যেই করেছি, এক আর দুই নম্বর দিনে। মেশিনের সমস্ত ভৌত উপাদানের ভিতর মাইক্রোপ্রসেসরই সেই কেন্দ্রীয় সত্তা যা অন্য সমস্ত কিছুকে চালায়। অথচ সে নিজে সরাসরি কোনো একটা অঙ্গপ্রত্যঙ্গকেও ছুঁতে পায়না। সব যোগাযোগই হয় বাস মারফত। বরং এক কোর মেমরির বা র‍্যামের সঙ্গেই প্রায়-সরাসরি, প্রায়-প্রত্যক্ষ একটা যোগ থাকে প্রসেসরের। কারণ প্রোগ্রাম চালাতে গেলে প্রোগ্রামগুলোকে মেমরিতে থাকতেই হবে। মনে পড়ছে ডিএমএ-র কথা? যখন হার্ডডিস্ক থেকে কোনো তথ্য পড়ার দরকার পড়ে, প্রসেসর তখন বাস মারফত একটা অনুরোধ পাঠায় ডিস্কপাঠের। বাস এই অনুরোধটাকে পৌঁছে দেয় হার্ডডিস্ক কন্ট্রোলারের কাছে। পড়ার কাজ শেষ হয়ে গেলে, ডিস্ক কন্ট্রোলার আবার বাস মারফত এই সিগনাল প্রসেসরকে পাঠায় যে অনুরুদ্ধ তথ্য পড়া হয়ে গেছে এবং তথ্যটা এখন মেমরির এত নম্বর খোপে তোলা আছে। এক নম্বর দিনের সেই রেজিস্টার এবং মেমরি লেখার ছকগুলো মনে করুন। এবার সেই তথ্য মেমরি থেকে তুলে নেয় প্রসেসর, আবার সেই বাস মারফত। এবং তথ্য চটকানোর কেন্দ্রীয় কাজে রত হয় সেন্ট্রাল প্রসেসিং ইউনিট বা সিপিইউ। কিবোর্ড বা কনসোলের সঙ্গেও প্রসেসরের যোগাযোগ ঘটে ওই বাস মারফত।

৬।। কম্পিউটারে প্রাণ-প্রতিষ্ঠা — বুটলোডার

যখন কোনো একটাও প্রোগ্রাম চলছে না — স্ক্রিনে কোনো লেখা বা ছবি কিছু নেই — হার্ড ডিস্কের কাজ করার আলোটাও অফ, তখন কম্পিউটারটা আর কিছুই না — অনেকটা কাঁচ প্লাস্টিক ধাতু তার স্ক্রু এইসবের একটা দামী মণ্ড। তাতে প্রাণ-প্রতিষ্ঠা হয়, কম্পিউটারটা কম্পিউটার হয়ে ওঠে যখন সেটা বুট করে। তার প্রথম কাজই তখন একটা বিশেষ প্রোগ্রামকে চালানো — যাকে নিয়ে আমাদের এই আলোচনাটা শুরু হয়েছিল — সেই প্রোগ্রামটার নাম অপারেটিং সিস্টেম। চালু হয়ে যাওয়ার পরে, অপারেটিং সিস্টেমের উপর এবার ভার পড়ে হার্ডওয়ারের সব বিতিকিচ্ছিরি ব্যাপারকে মাথায় রাখা সামলে রাখা নিয়ন্ত্রণ করা। যাতে অন্য সব প্রোগ্রাম হার্ডওয়ার ব্যবহার করতে

পারে। অপারেটিং সিস্টেমকে জাগিয়ে তোলার এই প্রক্রিয়াটারই অন্য নাম বুটিং। 'বুট' (boot) কথাটা এসেছে বুট মানে জুতো থেকে। বুট-স্ট্র্যাপিং-কে ছোট করে বুট। মানে, কাজটা কি শক্ত ভাবুন, নিজের জুতোর ফিতের ধরে নিজেকে টেনে তোলে, জ্যান্ত কাজের ভিতর নিয়ে আসে অপারেটিং সিস্টেম। আর শেষবার কম্পিউটার যখন অফ করে দিয়েছিলাম, তখন কোর বা র‍্যাম থেকে বাস থেকে সব তথ্য আর আদেশই বেরিয়ে ফুরিয়ে ভ্যানিশ করে গেছিল। শুধু রমে স্থায়ীভাবে লিখে রাখা সামান্য পরিমাণ অত্যাবশ্যক তথ্যটুকু ছাড়া মেশিনে আর কিছুই ছিলনা। খেয়াল রাখবেন, অফ-করা অবস্থায় একটা কম্পিউটারে কিন্তু কোনো অপারেটিং সিস্টেমই নেই। আপনার মেশিনে গ্নু-লিনাক্স ওএস ইনস্টল করলেন মানে, প্রয়োজনীয় ফাইল এবং কনফিগারেশন মেশিনটায় বানিয়ে তুললেন, এখন থেকে বিদ্যুৎ পাঠালেই সেখানে একটা গ্নু-লিনাক্স ওএস গজিয়ে উঠবে। সুইচ টিপে মেশিনে বিদ্যুৎ অন করার পর, সেই ওসটাই উঠতে থাকে, মানে মেশিন বা সিস্টেম বুট করতে থাকে। এই ওএস-নেই থেকে ওএস-আছে হওয়ার গোটা কার্যধারাটা বেশ জটিল। অনেকগুলো ধাপ আছে সেখানে, অনেক অনেক ধাপে অনেক অনেক আদেশ মেনে এগিয়ে তবে বুটপদ্ধতিটা সমাপ্ত হয়। এই কার্যধারার প্রাথমিকতম হদিশটা বুটমুহূর্তে মেশিনকে শেখায় বায়োস (BIOS — **B**asic-**I**nput-**O**utput-**S**ystem) — মানে অত্যন্ত প্রারম্ভিক কিছু নির্দেশ যা ভরা থাকে রম চিপের শরীরেই, যা মুছে যায়না, গেলে আর বুট করাই যেতনা। এই বায়োসে লিখে রাখা প্রাথমিকতম সংক্ষিপ্ত আদেশগুলো পালন করার মধ্যে দিয়েই বুটপ্রক্রিয়া শুরু হয়।

বায়োস সেই বুটমান ক্রমগজায়মান অপারেটিং সিস্টেমকে বলে দেয়, বাছা তুমি অমুক হার্ডডিস্কের অমুক জায়গা থেকে তোমার বুটপ্রক্রিয়ার পরবর্তী আদেশমালা পড়তে শুরু করো। সচরাচর ওই জায়গাটা হল মেশিনের প্রথম হার্ডডিস্কের প্রথম মানে সবচেয়ে কম-সংখ্যক সেক্টর। এই সেক্টরে পৌঁছনোর পর ওএস পাবে তার পরবর্তী ধাপের মন্ত্র। সিদ্ধি আসতে এখনো ঢের দেরি। এই মন্ত্রের নাম বুট-লোডার। বুট লোডার নানা রকম হয়। গ্নু-লিনাক্সে বুট লোডার মূলত গ্রাব (GRUB — **G**rand-**U**nified-**B**oot-**L**oader) আর লিলো (LILO — **Li**nux-**Lo**ader)। এছাড়া লোডলিন দিয়ে ডস বা উইন্ডোজ মারফতও বুট করানো যায়। আমরা এখানে আমাদের আশু আলোচনাতে লিলোকেই আনব, তার সোচ্চার কারণটা এই যে লিলোটা বোধহয় একটু কম জটিল। আর জনান্তিকে বলার কারণটা এই যে, আমি গ্রাব জানিনা। দু-একবার ঠেকায় পড়ে ব্যবহার করতে হয়েছে, সেটা একদম ধরো-পড়ো-করে-যাও নীতিতে, কিছুই না-বুঝে। গ্নু-লিনাক্সে প্রথম দিকে এটাই বোধহয় সবচেয়ে জনপ্রিয় এবং সফলতম আত্মশিক্ষানীতি।

এবার, প্রাণ-প্রতিষ্ঠার দ্বিতীয় ধাপ মানে লিলো সিনে এসে গেল মানেই, মেশিনের এবার কাজ হল ওই লিলোর আভ্যন্তরীন আদেশমালা পড়ে ফেলা এবং লিলোর গোটা তথ্যটাকে মেমরিতে তুলে ফেলা। এখন থেকে বুটপ্রক্রিয়ার গোটা দায়িত্বটাই লিলো নামের ওই বুট-লোডারের ঘাড়ে। সে এবার একটু একটু করে, ধাপে ধাপে, অপারেটিং সিস্টেমের অবশিষ্ট অংশটাকে ক্রিয়াশীল করতে শুরু করবে, জাগিয়ে তুলতে থাকবে। ক্রমে, তার কাজ সমাপিত হলে, বুটপ্রক্রিয়া শেষ হবে। বুটপ্রক্রিয়া ছাড়িয়ে সত্যিকারের জ্যান্ত একটা অপারেটিং সিস্টেম কাজ শুরু করবে কম্পিউটারে।

এই কাজটা বুটলোডার করবে স্টেপ-বাই-স্টেপ। প্রথমে সে একটা কারনেল খুঁজে বার করবে এবং সেটাকে তুলে নেবে মেমরিতে। পরে যখন আমরা লাইন ধরে ধরে বুটলোডার বুঝব, তখন দেখব, বুটলোডারের মূল কাজ এটাই, সঠিক উপযোগী কারনেলটা কোথায় আছে সেটা জানিয়ে দেওয়া। সেই কারনেল এবার চালু হবে, এবং নিজের কাজ শুরু করে দেবে। এইখানে এসে বুটলোডারের কাজ শেষ। বুটলোডারে দেওয়া থাকতে পারে একাধিক সম্ভাব্য কারনেলের হদিশ। ধরুন একটা মেশিনে একই সঙ্গে উইন্ডোজ আছে এবং গ্নু-লিনাক্স আছে দুটো, সুজে এবং স্ল্যাকওয়ার। মেশিন অন হওয়ার পর প্রথম ফুটে উঠবে লিলোর স্ক্রিন। লিলো জানতে চাইবে, আমরা কোন ওএস-এ বুট করতে চাই? উইন্ডোজ না সুজে না স্ল্যাকওয়ার। যদি আমরা উইন্ডোজ বলি, লিলো নিয়ন্ত্রনটা সরাসরি চলে যেতে দেবে উইন্ডোজ লোডারের হাতে, যা প্রথম হার্ডডিস্কের প্রথম পার্টিশনের গোড়াতেই লেখা থাকে — উইনডোজে এরকমই প্রথা। আর গ্নু-লিনাক্স যে দুটো অপারেটিং সিস্টেম আছে, সুজে বা স্ল্যাকওয়ার, তার দুজনেরই কার্নেল কোথায় আছে, এবং কোন পার্টিশনকে কোন সিস্টেমের মূল বলে ধরতে হবে — সেটাও লিপিবদ্ধ আছে লিলোর কাছে। ধরুন আমরা সুজেতে যেতে চাই, সেক্ষেত্রে লিলো সুজের কার্নেলটা যেখানে রাখা আছে সেটা পড়বে এবং চলে যাবে সুজে সিস্টেমের মূল বা রুট বলে যে পার্টিশন দেখানো আছে, সেইখানে। সেখানেই কাজ শুরু করবে।

লিলো কাজ করে লিলোর কনফিগারেশন বা খুঁটিনাটি-লেখা ফাইল দিয়ে। যার নাম লিলো-কনফ। এই কনফিগারেশন ফাইল, তাদের হদিশ, তাদের গঠন — এসব নিয়ে আমাদের পরে খুবই বিশদ করে জানতে হবে। এখন একটা আলগা আন্দাজ বানানোর চেষ্টা হিশেবে এই লিলো-কনফ (lilo.conf) ফাইলটার গঠনটা দেখুন। মনে রাখবেন, যেহেতু প্রায় কিছু না-জেনেই আপনাকে ফাইলটা পড়তে হচ্ছে, কিছুটা গুলিয়ে যাবেই। কিছু করার নেই। জাস্ট একটু আন্দাজ রেখে যেতে চাইছি। না-বোঝাটা নিয়ে মাথা ঘামাবেন না। পড়ে যান। পরিচিতিটুকুই আপাতত দরকার আমাদের।

'lilo.conf' ফাইলটায় চারটে অংশ আছে। প্রথমে একটা জেনেরাল বা সাধারণ অংশ। একদম উপরের প্রথম দশ লাইন। তারপর এক এক লাইন করে সাদা জায়গা ছেড়ে দিয়ে তিনটে অংশ — সুজে (SuSE), স্ল্যাকওয়ার (Slackware) আর উইনডোজ (Windows)। প্রথম দুটো অংশ সুজে আর স্ল্যাকওয়ার এদের দুজনেরই কারনেল সংক্রান্ত খুঁটিনাটি সিস্টেম কোথা থেকে পাবে তার হদিশ। গ্নু-লিনাক্স সিস্টেমে হার্ডডিস্ক নামকরণের প্রথা অনুযায়ী এই মেশিনের দুটো হার্ডডিস্কের নাম হল 'hda' আর 'hdb', হার্ডডিস্ক এ আর বি, ধরুন। আর গ্নু-লিনাক্সে, আমরা আগেই বলেছি, সব কিছুই এক একটা ফাইল। হার্ডডিস্ক মানে ডিভাইস ফাইল, তারা আছে /dev ডিরেক্টরিতে। এই সিস্টেমে দুটো হার্ডডিস্ক মানে দুখানা ফাইল। তাদের ঠিকানা '/dev/hda' আর '/dev/hdb'। প্রত্যেকটা হার্ডডিস্কে আবার একাধিক করে পার্টিশন আছে। যাদের নম্বর ১, ২, ৩, ৪ করে। এই খ্যাঁচগুলো পরে শিখব আমরা। আপাতত এটুকু বুঝুন, '/dev/hda1' মানে প্রথম হার্ডডিস্কের প্রথম পার্টিশন, '/dev/hda6' মানে প্রথম হার্ডডিস্কের ছয় নম্বর পার্টিশন, '/dev/hdb3' মানে দ্বিতীয় হার্ডডিস্কের তিন নম্বর পার্টিশন।

লিলো-র কনফিগারেশন ফাইল lilo.conf এর একটা উদাহরণ। নিজের এবং নিজের মনের স্বাস্থ্যের কথা ভেবে এর গোটাটা এখনি বোঝার চেষ্টা করবেন না, প্লিজ। এখানে তিনটে আলাদা অপারেটিং সিস্টেম দিয়ে এই মেশিনে বুট করার ব্যবস্থা করা আছে — সুজে, স্ল্যাকওয়ার এবং উইনডোজ। মোটামুটি গড়নটার একটা আন্দাজ পাওয়ার চেষ্টা করুন।

```
boot = /dev/hda
change-rules
reset
default = SuSE
lba32
menu-scheme = Wg:kw:Wg:Wg
message = /boot/message
prompt
read-only
timeout = 150

image = /boot/vmlinuz
    label = SuSE
    append = "hdc=ide-scsi splash=0"
    initrd = /boot/initrd
    root = /dev/hdb3
    vga = 789

image = /mnt/slackware/boot/vmlinuz
    label = SlackWare
    append = "hdc=ide-scsi"
    root = /dev/hda6
    vga = 789


other = /dev/hda1
    label = windows
```

তিনটে আলাদা ওএস-এর তিনটে ব্লক আছে পরে, তার আগে সেই অংশটা যেটা প্রত্যেকের বেলাতেই সত্যি। পরে এই লাইনগুলোকে ভালো করে বোঝার চেষ্টা করব আমরা।

'lilo.conf' ফাইলটার প্রথম লাইনটা দেখুন, 'boot = /dev/hda', এর মানে বুটসেক্টরটা রয়েছে প্রথম হার্ডডিস্কের গোড়াতেই। তারপরের আটটা লাইন আপাতত ছেড়ে দিন। দশ নম্বর লাইন, 'timeout = 150', এর মানে সেকেন্ডের এক দশমাংশের এককে ১৫০ একক, অর্থাৎ পনেরো সেকেন্ড সময় দেবে লিলো। এই পনেরো সেকেন্ড দাঁড়িয়ে

থাকার পরেও যদি আপনি কোনো একটা বিশেষ ওএস-কে নির্বাচন করে না-দেন, তাহলে ও নিজেই বুট করে যাবে ডিফল্ট ওএস-এ। এখানে ডিফল্টটা হল সুজে। দেখুন, চার নম্বর লাইনে আছে, ‘default = SuSE’। আপনি পছন্দ করে তিনটের যে কোনোটাই দিতে পারেন। যেটাকেই আপনি পছন্দ করুন, সেই ওএস-এর কারনেলের ঠিকানা দেওয়া আছে তার নিজের অংশে। তিনটে ব্লক আছে দেখুন, সুজে স্ল্যাকওয়ার আর উইনডোজের। ছয়, পাঁচ আর দুই লাইন করে।

যেমন সুজে অংশের প্রথম লাইন দেখুন, ‘image = /boot/vmlinuz’। এর মানে, ওই ‘vmlinuz’ নামের ইমেজ বা সুজের কারনেল ফাইলটা রাখা আছে ‘/boot’ নামের ডিরেক্টরিতে। এই ডিরেক্টরিকে কোথায় খোঁজা হবে? সুজে সিস্টেমের শিকড়ে, রুট ডিরেক্টরিতে। তার ঠিকানা কী? সুজে ব্লকের পাঁচ নম্বর লাইনে দেখুন, ‘root = /dev/hdb3’। জ্যামিতিতে যেরকম আমরা অরিজিন শিফট করে অন্য জায়গায় নিয়ে যাই এখানেও তেমনি একটা ঘটনা ঘটছে। বুট করার সময়, বুট লোডার পড়ার সময়, প্রথম জেনেরাল অংশে দেখুন, শিকড় দেখাচ্ছিল প্রথম হার্ডডিস্ককে। আগেই বললাম না, প্রথম হার্ডডিস্কের গোড়াতেই লেখা থাকে বুটলোডার, পরে এর ভৌত ব্যাপারটা বুঝব আমরা, হার্ডডিস্কের গঠনের আলোচনায়। আর সুজের কারনেলের প্রসঙ্গে দেখুন, শিকড়টাকে সরিয়ে নিয়ে যাওয়া হচ্ছে দ্বিতীয় হার্ডডিস্কের তিন নম্বর পার্টিশনে। ‘/dev/hdb3’ বা দু নম্বর হার্ডডিস্কের তিন নম্বর পার্টিশনটাই শিকড় বলে ভাবা হবে, সেখানে গিয়ে সিস্টেম তার প্রয়োজনীয় ফাইলগুলো খুঁজবে। একই রকম একটা সরিয়ে নিয়ে যাওয়া ঘটছে স্ল্যাকওয়ারের বেলাতেও। শিকড়টা বদলে যাচ্ছে ‘/dev/hda6’ বা প্রথম হার্ডডিস্কের ছয় নম্বর পার্টিশনে, কারণ সেটাই স্ল্যাকওয়ারের রুট। রুট পার্টিশন কী বস্তু সেটা আমরা ভালো করে জানব সাত নম্বর দিনে গিয়ে।

সুজে অংশে এর পরের লাইন দেখুন, ‘initrd = /boot/initrd’। ‘initrd’ শব্দটা হল ‘initialized-ramdisk’ শব্দবন্ধের সংক্ষিপ্ত আকার। আট নম্বর দিনে গিয়ে আর একটু ভালো করে বুঝব আমরা। র‌্যামডিস্ক বলতে র‌্যামের একটা অংশকে আলাদা একটা ডিস্ক আকারে ধরে নিচ্ছে মেশিন, এবং সেই র‌্যামডিস্কটা ইনিশিয়ালাইজ করার কথা বলা হচ্ছে, মানে বুটলোডার যাতে এই র‌্যামডিস্কে কারনেলের চালু হওয়ার জন্যে প্রয়োজনীয় নানা ডিভাইস ড্রাইভার মডিউল তুলে নেয়, যা সিস্টেম চালু হওয়ার জন্যে দরকার পড়বে। কারনেল এই র‌্যামডিস্ক থেকে তার প্রয়োজনীয় ড্রাইভারগুলোকে পড়ে নেবে। এই লাইনটা কারনেলকে বলে দেয় কোথায় সে এই র‌্যামডিস্কটাকে পাবে তার ঠিকানা, মানে ‘/boot/initrd’ নামের ফাইল। দেখুন, এই লাইনটা স্ল্যাকওয়ার অংশে নেই, কারণ স্ল্যাকওয়ার কারনেল অন্য ভাবে কাজ করে। সুজে অংশে এর পরের লাইন হল ‘vga = 789’। এই লাইনটা বলে দেয়, কী রকম স্ক্রিনে কী সাইজের অক্ষর দিয়ে সে কমান্ড প্রম্পট ফুটিয়ে তুলবে তার হদিশ। এর নানা মান হতে পারে। যেমন, ৭৭১ মানে হল ৮০০×৬০০ পিক্সেল আকারের স্ক্রিনে আট বিটে ২৫৬ রঙের ডিসপ্লে। এখানে যে মানটা আছে, সেই ৭৮৯ মানে হল ওই একই সাইজের স্ক্রিনে ২৪ বিটে ১৬ মিলিয়ন রঙের ডিসপ্লে। মানটা ৭৯১ হলে, তার মানে হত ১০২৪×৭৬৮ পিক্সেল আকারের স্ক্রিনে ১৬ বিটে ৬৫ হাজার রঙের ডিসপ্লে। এইরকম। এর কোনটা করা যাবে, সেটা নির্ভর করে হার্ডওয়ারের উপর, র‌্যামের উপর, ভিডিও র‌্যামের উপর। এই জায়গাটায় বেশ ইন্টারেস্টিং অনেককিছু আছে, কিন্তু সেগুলো আমাদের এই পাঠমালায় আসবে না, ওগুলো ভালো করে বুঝতে হত এক্স-উইনডোজ বা গুই জানতে চাইলে। সেটা তো আমরা প্রথমেই আমাদের আলোচনা থেকে বাদ দিয়ে দিয়েছি।

‘lilo.conf’ ফাইলের সুজে বা স্ল্যাকওয়ার ব্লক দুটোয় দেখুন, খুব সামান্যই তফাত। শুধু ওই র‌্যামডিস্কের লাইনটা বাড়তি আছে সুজেতে। এছাড়া কাঠামোটা দুটোরই এক। উইনডোজ অংশে দেখুন এসব খুঁটিনাটি কিছুই নেই, কারণ উইনডোজ ওএস-এর হালহকিকত এবং বুট করার তরিকা দেওয়া আছে প্রথম হার্ডডিস্কের প্রথম সেক্টরেই। সেখান থেকে, সেই অনুযায়ী, পরপর সিস্টেম পড়ে নেবে, এবং লোড করবে। তার ব্যাকরণ গ্নু-লিনাক্স থেকে একদম আলাদা। এই যে লিলো কনফিগারেশন ফাইলের কিছু লাইনকে আমরা একদম ধরলামই না, এদের কিছু আলোচনা পরে আসবে।

তাহলে দেখুন, একাধিক সিস্টেমের একাধিক কারনেল একই সঙ্গে নানা হার্ডডিস্কের নানা পার্টিশনের নানা জায়গায় আগে থেকেই ভরা আছে। শুধু যে একাধিক সিস্টেমের একাধিক কারনেল তা নয় কিন্তু। একই সিস্টেম একাধিক কারনেল দিয়ে লোড করা যেতে পারে। ধরুন সুজে সিস্টেমে এমনিতে দেওয়া কারনেলে আমি আমার সিস্টেম বুট করি। এবার আমার প্রয়োজন হল, অন্য কিছু মডিউল কারনেলে ঢোকানোর। ধরুন আমি আরো দ্রুতগতি করার জন্যে

স্কাসি ব্যবস্থা করলাম মেশিনে। স্কাসি হার্ডডিস্ক লাগালাম আইডিই হার্ডডিস্কের বদলে। এবার আমার চালু কারনেলটায় যদি স্কাসি ড্রাইভার এমনিতে না-থাকে, তাহলে সেই ড্রাইভারটা সঙ্গে দিয়ে আমি একটা নতুন কারনেল দিয়ে আমার সিস্টেম লোড করতে পারি। বা, নতুন কারনেল সংস্করণ বেরোলো, সেটাকে কম্পাইল করে নেওয়া যেতে পারে নিজের সিস্টেমে। জিসিসি কম্পাইলার তো দেওয়াই থাকে গ্নু-লিনাক্স বাক্সে। যেমন এখন নতুন কারনেল হল ২.৬। আমাদের কোলকাতা লাগের সায়মিন্দু সহ অনেকেই বেশ মাসতিনেক হয়ে গেল ২.৬ কারনেল নামিয়ে নিজের মেশিনে কম্পাইল করে নিয়েছে। আমারটায় পুরোনো কারনেলই চলছে। এক, আমার অত আপটুডেট ড্রাইভার প্রয়োজনও পড়েনা, আর ওদের মত করে এইসব হার্ডওয়ার জটিলতা আমি আদৌ বুঝিনা। দু-একবার কারনেল কম্পাইল করেছি, কিন্তু সে বেশ লাগে-তুক-না-লাগে-তাক মেথডে। যাই হোক, যা বলছিলাম, একই সিস্টেমেরও একাধিক কারনেল থাকতে পারে। যে কোনো একবার শুধু একটা কারনেল দিয়েই বুট করা যাবে, যে কারনেলটা আপনি পছন্দ করে দেবেন লিলোকে।

বিদ্যুৎ অন করার পর, প্রাথমিক কিছু হার্ডওয়ার সংক্রান্ত লাইন স্ক্রিন জুড়ে চলে যাওয়ার পর, প্রথম যেটা ফুটে উঠবে সেটা লিলো স্ক্রিন, যেখানে লিলো তার সম্ভাব্য কারনেল কাম ওএস-গুলোর তালিকা ফুটিয়ে তুলবে। সেখান থেকে একটা কোনো কারনেলকে আপনি পছন্দ করে দেবেন। ‘lilo.conf’ ফাইল মোতাবেক লিলো ইনস্টল করে নিতে হয়। ইনস্টল করতে হয় ‘`lilo`’ কমান্ড দিয়ে, ‘/etc’ ডিরেক্টরিতে ‘lilo.conf’ ফাইলটা লিখে নেওয়ার পর। লিলো কী ভাবে ইনস্টল করা হবে, কী কী কারনেল রাখা হবে তার তালিকায়, সেইসব কনফিগারেশনটাই লেখা থাকে ‘lilo.conf’ ফাইলে — পরে এগুলো আরো বুঝব আমরা। লিলো ইনস্টল করার পর তৈরি হয় বুটকালীন কারনেল তালিকা ফুটে ওঠার এই স্ক্রিন। এই তালিকায় পরপর তিনটে ওএস-এর নাম ফুটে উঠবে, আমরা আমাদের একটু আগের উদাহরণ হিশেবে দেওয়া ‘lilo.conf’ ফাইলে সেই রকম কনফিগারেশনই দিয়েছি লিলোকে। এই তিনটে পরপর নাম হল, ‘SuSE’, ‘Slackware’, এবং ‘windows’। দেখুন, ‘lilo.conf’ ফাইলের তিনটে ব্লকে তিনটে আলাদা নাম দেওয়া আছে, ‘label = ’ লাইনে। মানে তিনটে ওএস-এর তিনটে কারনেলের হদিশ। যদি সুজের দুটো কারনেল থাকত তাহলে আবার এখানে পছন্দের সংখ্যা চারটে। এবার, এই লিলো স্ক্রিনে দেওয়া তালিকায় উর্দ্ধমুখী তীরচিহ্ন (↑) আর নিম্নমুখী তীরচিহ্ন (↓) দিয়ে উঠে নেমে কোনো একটা নির্দিষ্ট অপারেটিং সিস্টেমকে, মানে সেই সিস্টেমের কারনেলকে পছন্দ করে আমরা যেই এন্টার মারি, সেই কারনেলটা লোড হতে শুরু করে। মানে তার ঠিকানা থেকে তাকে মেমরিতে তুলে নিতে শুরু করে মেশিন। অল্প সময়ের জন্যে কনসোলের কালো পর্দায় পরপর কতকগুলো সাদা বিন্দু ফুটে উঠতে থাকে — মানে, লিলো তখন কারনেল লোড করছে। এখনকার বিদ্যুৎগতি সব প্রসেসরের কল্যাণে এই ডিসপ্লেটা প্রায় চোখের পলক ফেলতে না-ফেলতেই মিলিয়ে যায়।

যাইহোক, কারনেল মেমরিতে উঠে গেল, জ্যান্ত হয়ে উঠল, তার নিজের কাজ শুরু করে দিল — এইখানেই লিলোর কাজ শেষ। এখানে মনে হতে পারে, শুধু শুধু লিলোকে আনা কেন? কারনেল পছন্দ করা, পছন্দ করা কারনেলকে তার ঠিকানা থেকে পড়তে শুরু করা, তারপর সবার শেষে কারনেলটাকে জ্যান্ত করে তোলা — এই অব্দি দায়িত্বটা তো বায়োসের কাছে থাকলেই পারত? তাহলে লিলো বলে আলাদা কোনো একটা প্রোগ্রামকে চালু করার দরকার পড়ত না। এতে সমস্যা এই যে, বায়োসে কতটা স্মৃতি রাখা যাবে তার পরিমাণটা খুব সীমিত। সেই অর্থে কম্পিউটারের বায়োস জিনিষটা বেশ আকাঠ, বায়োসের কোডগুলো প্রথম লেখা হয়েছিল যখন, তখন কম্পিউটার কাজ করত ৮ বিটে, মানে ৮ বিট ছিল তার স্মৃতি নাড়াচাড়ার একক। আর তথ্য লেখাপড়ার কাজগুলো হত এখনকার তুলনায় হাফটিকিট ওয়ানফোর্থ টিকিট কি আরো কুচো কুচো হার্ডডিস্কে। এখন এই ৩২ পেরিয়ে ৬৪ বিটের দিকে ধাবমান মেশিনের যুগেও বায়োস হুবহু এক না-থাকলেও আমূল বদলে গেছে এমনটাও বলা যায়না। তাই এই বায়োসের পক্ষে আজকের দৈত্যাকার হার্ডডিস্কের এত কম অংশই মাথায় রাখা সম্ভব যে তার দ্বারা সরাসরি কারনেল লোডিং সম্ভব নয়। বায়োস আপগ্রেড করে, যার অন্য নাম বায়োস ফ্ল্যাশিং, মানে বায়োসের পুরোনো তথ্য মুছে নতুন তথ্য লিখে বায়োসের কিছু জায়গা বদলানো যায় ঠিকই, কিন্তু, প্রথমত, বায়োস ফ্ল্যাশিং বেশ ঝঞ্ঝাটের কাজ, আর দ্বিতীয়ত, ফ্ল্যাশ করে তো আর আমরা বায়োসে রাখা তথ্যের পরিমাণ বাড়াতে পারছি না। আর, বুট লোডারের সুবিধাটা এই যে, তথ্যটা যেহেতু লেখে হার্ডডিস্কে, যতখুশি জায়গা, তাই, একসঙ্গে মেশিনে যতখুশি অপারেটিং সিস্টেম রেখে তার যে কোনোটাতে ঢুকে যেতে পারার সুযোগ।

৭।। কারনেল নিজের জমি বুঝে নিচ্ছে — ইনিট

কারনেল কাজ করতে শুরু করার পরই প্রথম যাতে হাত দেয় তা হল হার্ডওয়ারটাকে বুঝে নেওয়া মিলিয়ে নেওয়া। যাতে এরপরেই প্রোগ্রাম চালানোয় হাত দেওয়া যায়। এই হার্ডওয়ার খোঁজার সময় কারনেল যে গোটা মেশিনের সমস্ত কিছু খুঁজে দেখতে দেখতে যাচ্ছে তা কিন্তু নয়। কতকগুলো বিশেষ জায়গা আছে, যেমন ইনপুট-আউটপুট সংযোগবিন্দু বা আইও-পোর্টগুলো — বিশেষ কিছু বাস-রুটের ঠিকানা যেখানে কন্ট্রোলার কার্ডগুলো খাপ পেতে বসে থাকে — কখন তাদের আদেশ পাঠানো হবে কাজ করার — ইত্যাদি। মানে, কারনেল যে এলোমেলোভাবে হাতড়ে চলে তা নয়, কারনেলের মধ্যেই পর্যাপ্ত পরিমাণে এই তথ্য ভরা আছে যে কোথায় হাতড়ালে কী তার খুঁজে পাওয়ার কথা, আর ঠিক কী আকারে খোঁজ করতে চাইলে তবেই তার এই সন্ধানে সাড়া দেবে কার্ডগুলো। ফিজায় ঋত্বিকের মত বারবার নতুন করে খুঁজতে হবে, নাকি, রোড-এর মনোজ বাজপেয়ীর মত খোঁজা শেষ করে নিশ্চিন্ত হওয়া মাত্রই হঠাৎ করে উদয় হবে — এসব কারনেল আগে থেকেই জানে, তাকে চমকানো খুব শক্ত।

এই বিশেষ রকমে শিক্ষিত খোঁজাখুঁজিরই টেকনিকাল নাম অটোপ্রোবিং। স্বসন্ধান বলা যায়। গ্নু-লিনাক্স একটা ব্যবস্থায়, বুট করা শুরু করার পরেই, কনসোলের পর্দায় যে রাশি রাশি লাইন সাঁই সাঁই করে নিচ থেকে উপরে চলে যায় — সেসব কারনেলের এই অটোপ্রোবিং-এর ফলাফল। আইও পোর্টগুলো বাজিয়ে যাচাই করে যে বুঝে নিচ্ছে — কী কী রসদ এইমুহূর্তে সে পেতে পারে মেশিনের মধ্যে — পরে প্রোগ্রাম চালাতে গিয়ে কোনো ঘাপলা যাতে না-ঘটে। আর এক একটা করে উপাদান খুঁজে পাচ্ছে এবং নিজেকে সেইরকম করে প্রস্তুত করে নিচ্ছে কারনেল, মেশিনের সাথে নিজেকে একদম খাপে খাপে মিলিয়ে নিচ্ছে। এই কাজে গ্নু-লিনাক্স কারনেল বেজায় দড়। অন্য সমস্ত ধরনের ইউনিক্সের চেয়ে অনেকটা দক্ষ, আর ডস বা উইনডোজের কথা তো ছেড়েই দিন। অনেক পুরোনো লিনাক্সীই মনে করে যে, নানা ধরনের আবদ্ধ এবং উন্মুক্ত ইউনিক্সের মধ্যে থেকে গ্নু-লিনাক্স যে নিজের আলাদা জায়গা বানাতে শুরু করেছিল তার একটা বড় কারণ এত উচ্চস্তরের বুটকালীন স্বসন্ধান। এই উমদা অটোপ্রোবিং-এর কারণে গ্নু-লিনাক্স ইনস্টল করার ঝামেলা অনেকটা কমে যায়। নিজেই কারনেল খুঁজে নেয় ভৌত উপাদানগুলো এবং তাদের ড্রাইভার নিজেই লোড করে নেয়, কিছুই প্রায় তাকে আলাদা করে বলতে হয়না।

কিন্তু গোটা কারনেলটা মেমরিতে তুলে নেওয়া এবং চালানো মাত্রই কিন্তু বুটপ্রক্রিয়া শেষ নয়। অ্যাকচুয়ালি এটা প্রথম পদক্ষেপটা সমাপ্ত হল মাত্র। এই প্রথম স্টেপের শেষে কারনেল যে দায়িত্ব, মানে বুটপ্রক্রিয়ার নেতৃত্ব হাতে পেয়েছিল, সেই পতাকা এবার তুলে দেবে ইনিট-এর হাতে, যে ইনিট ফের শুরু করবে নতুন দিনের শুরুতে মেশিনের ঘরগেরস্থালি ঠিকঠাক ভালো করে গুছিয়ে নেওয়ার অনেকগুলো কাজ। ইনিট (init) আর একটা প্রোগ্রাম, ইনিট হল ইনিশিয়ালাইজেশন। একটা সিস্টেম প্রোগ্রাম, তাই রুট ছাড়া কেউ হাত দিতে পারেনা এই প্রোগ্রামে বা তার কনফিগারেশন ফাইলে। ঠিক লিলো প্রোগ্রামের যেমন কনফিগারেশন ফাইল ছিল 'lilo.conf', তেমনি ইনিট প্রোগ্রামেরও একটা কনফিগারেশন ফাইল থাকে, তার নাম 'inittab'। সেটাও 'lilo.conf' ফাইলেরই মত থাকে সিস্টেমের '/etc' ডিরেক্টরিতে। পরে আমাদের খুবই ভালো করে জানতে হবে এগুলো। আর '/etc' ডিরেক্টরিটাকেও আলাদা ভাবে জানব আমরা নয় নম্বর দিনের ১.৪ নম্বর সেকশনে গিয়ে। আর আগেই তো বলেছি, এখুনি যদি কৌতূহল মেটাতে চান, যা খুবই ভালো অভ্যেশ, নিজের গ্নু-লিনাক্স সিস্টেমে দাঁড়িয়ে ম্যান (man) কমান্ড ব্যবহার করুন। আমি যেভাবে চেয়েছি তাতে এই পাঠমালাটা সত্যিই অর্থপূর্ণ হয়ে উঠবে যখন নিয়ন্ত্রণটা আমার হাত থেকে আপনি নিজের হাতে নিয়ে নেবেন। প্রথমদিকে এটা হবে খুব ছোট ছোট জার্নি। পরে সেটা বাড়তে থাকবে।

এই ইনিট বা ইনিশিয়ালাইজেশন প্রক্রিয়া হল সিস্টেম বুট করার পর, মানে বুটলোডারের হাত থেকে কারনেলের হাতে নিয়ন্ত্রণটা চলে যাওয়ার পর, কম্পিউটারে চলা প্রথম প্রক্রিয়া বা প্রসেস। ইনিট প্রোগ্রামের প্রসেস আইডি ১। আমরা এক আর দু-নম্বর দিনে প্রসেস নিয়ে আলোচনা করেছি, একটা প্রোগ্রাম যেই চলে সেটা একটা প্রসেস হয়ে ওঠে। প্রত্যেকটা প্রসেসেরই একটা করে আইডি বা আইডেন্টিটি বা নম্বর থাকে এবং থাকে বংশপরিচয়। মানে, উপরদিকে থাকে মাতাপিতা-প্রসেস, যে প্রসেসের থেকে এই প্রসেস উদ্ভূত হয়েছে, আর নিচের দিকে ছানাপোনা-প্রসেস, মানে এই প্রসেস থেকে যে প্রসেসরা উদ্ভূত হয়েছে। একমাত্র এই ইনিট ছাড়া। ইনিট হলেন স্বয়ম্ভূ, কোনো মাতাপিতা নেই, যদিও, রাবণের বংশের মত ছানাপোনা আছে। একটা চলমান মেশিনে গোটা গ্নু-লিনাক্স সিস্টেম জুড়ে কাজ করতে থাকা সমস্ত প্রসেস-ই এই ইনিটের বংশধর, তাই এর আইডি ১।

পরে আমরা ইনিট প্রোগ্রাম এবং তার কনফিগারেশন ফাইল 'inittab' এগুলোকে অনেক বিশদে জানব। 'inittab' ফাইলের লাইন ধরে ধরে। আপাতত এটুকু মাথায় রাখুন, ইনিট কাজ করে রানলেভেল (runlevel) বলে একটা জিনিষ দিয়ে, যার মান এক থেকে ছয় অব্দি হতে পারে। 'init 1', 'init 2' থেকে 'init 6' অব্দি যে কোনো কমান্ড দিয়ে সেই সেই রানলেভেলকে ডেকে আনা যায়। '/etc' ডিরেক্টরির 'inittab' ফাইল, মানে ইনিটের কনফিগারেশন ফাইল থেকে কয়েকটা লাইন তুলে দিই। এই লাইনগুলো ফাইলে রাখা আছে ইনিট প্রক্রিয়াটা আমাদের বোঝার স্বার্থেই। যার প্রমাণ লাইনের গোড়ায় ওই '#' চিহ্নটা। এই চিহ্নটা লাইনের গোড়ায় থাকলে সেই লাইনগুলোকে বলে কমেন্ট লাইন। মেশিন এই লাইনগুলো পড়েনা, আমরা পড়ি, যদি পড়তে চাই। তবে মাথায় রাখবেন এটা সুজে সিস্টেমের ইনিটট্যাব ফাইল থেকে তোলা। গ্নু-লিনাক্সেরই একটা ডিস্ট্রিবিউশন থেকে আর একটা ডিস্ট্রিবিউশনের সিস্টেমে এই রানলেভেলগুলো অনেকসময় একটু একটু পৃথক হয়। লাইনগুলোয় অনেক জিনিষই আপাতত বুঝতে পারবেন না। অল্প জেনে বেশি বোঝার প্র্যাকটিস শুরু করে দিন। তাতে মাঝে মাঝে একটু কেলেঙ্কারি ঘটতে পারে, কিন্তু মোটের উপর বোঝাটা বাড়বে। রানলেভেল ০ মানে সিস্টেম বন্ধ করছেন, আর ৬ মানে নতুন করে বুট করছেন। রানলেভেল ১ মানে যেখানে একটাই মাত্র ব্যবহারকারী থাকবে সিস্টেমে, সেই রুট, সেই সাধারণ ইউজার। রানলেভেল ২ মানে একাধিক ইউজার থাকতে পারে, কিন্তু নেটওয়ার্ক নেই। রানলেভেল ৩ মানে মাল্টি-ইউজার এবং নেটওয়ার্ক দুই-ই পুরোপুরি আছে। রানলেভেল ৫ মানে এই রানলেভেল ৩ গোটাটাই, আর তার সঙ্গে এক্সডিএম (xdm — **x-d**isplay-**m**anager), মানে এক্স-উইনডোজ বা গুই ব্যবহার করার সুযোগ। আমি নিজে যেটা পছন্দ করি, সেটা হল রানলেভেল ৩ দিয়ে সিস্টেম বুট করা, তারপর ইচ্ছেমতন এক্স-উইনডোজে ঢোকা বা বেরোনো। আর রানলেভেল ৫ মানে যেখানে প্রথমেই আমি গুই দিয়ে ঢুকছি, এক্স-উইনডোজের বাইরে যেতেই পারছিনা। ছবি আর ইঁদুর — জিনা ইয়াহা মরনা ইয়াহা ইসকা সিওয়ায়ে যানা কাহা। আর, রানলেভেল ৪ সুজেতে ব্যবহারই হয়নি।

```
# runlevel 0  is  System halt   (Do not use this for initdefault!)
# runlevel 1  is  Single user mode
# runlevel 2  is  Local multiuser without remote network (e.g. NFS)
# runlevel 3  is  Full multiuser with network
# runlevel 4  is  Not used
# runlevel 5  is  Full multiuser with network and xdm
# runlevel 6  is  System reboot (Do not use this for initdefault!)
```

এক থেকে ছয় — কোন মানটা আমি ব্যবহার করতে চাই আমার সিস্টেমের ইনিটের রানলেভেল হিশেবে, সেটা দিয়ে দিতে হয় সিস্টেমকে, 'inittab' ফাইল বদলে। এটা সিস্টেম প্রোগ্রামের এলাকা বলে রুট ছাড়া আর কেউ হাত দিতে পারেনা এই ফাইলে। পরে আমরা দেখব, ইনিট প্রোগ্রামের কাজ হল গেটি (getty — **get-tty**) বলে আর একটা প্রোগ্রাম চালু করা। গেটি ব্যবহারকারীদের হয়ে 'tty' বা টেলিটার্মিনাল পেয়ে দেয়। চার নম্বর দিনের বহু-ব্যবহারকারীর সিস্টেম তৈরি হওয়ার আলোচনাটা মনে করুন। অনেকগুলো টেলিটার্মিনাল, তারা প্রত্যেকটাই যুক্ত মূল একটা সিপিইউ-র সঙ্গে। গ্নু-লিনাক্স তথা যে কোনো ইউনিক্সেই চিন্তার মূল মডেল এটাই। এবার, কোন টেলিটার্মিনাল গেটি পেয়ে দেবে ব্যবহারকারীর জন্যে, ব্যবহারকারী সেখানে কী কী সুযোগ এবং অধিকার পাবে, সেটাই ঠিক করা হয় রানলেভেল দিয়ে। 'inittab' ফাইলে যে রানলেভেল ঠিক করে দিচ্ছি আমরা, সেই মত সুযোগ ব্যবহারকারীকে দেবে গেটি। সাত নম্বর দিনে অধিকার এবং মালিকানার ব্যাপারটা স্পষ্ট হওয়ার আগে অব্দি এর গোটাটা বুঝতে পারবেন না। আলগা বোঝা দিয়েই কাজ চালাতে থাকুন, ছেড়ে দেবেন না। একসময় দেখবেন গোটা ব্যাপারটাই ধরে ফেলেছেন।

'/etc' ডিরেক্টরিতে 'inittab' ফাইলে ডিফল্ট ইনিট রান-লেভেল ঠিক করে দেওয়ার একটা লাইন থাকে। সেটাকে বদলে আমরা আমাদের সিস্টেমের ডিফল্ট রানলেভেল বদলাই। ইনিটট্যাব-এ লিপিবদ্ধ আলাদা আলাদা রান-লেভেল ২ বা ৩ বা ৫ অনুযায়ী কম্পিউটার ঢোকে মাল্টি-ইউজার মোডে বা নেটওয়ার্ক মোডে বা গুই মোডে। আগেই তো বললাম, এর এমন কোনো নিয়ম নেই যা সব ডিস্ট্রিবিউশনেই একইভাবে প্রযোজ্য। সুজেতে যা তা স্ল্যাকওয়্যারে নয়। শুধু ০ আর ৬ রানলেভেল নির্দিষ্ট থাকে কম্পিউটার শাটডাউন বা অফ করে দেওয়ার সঙ্গে। ইনিটট্যাব ফাইলে আমরা আলাদা করে যে কোনো একটা রানলেভেলেই ঢোকার কথা বলে দিতে পারি।

ইনিট এবার চালায় কিছু স্ক্রিপ্ট। স্ক্রিপ্ট কাকে বলে তার একটা হালকা আন্দাজ দিয়েছি আমরা দিয়েছি, সেটা কিন্তু শোকেস বা তারও কম। গুদামে আরো প্রচুর আছে। দশ নম্বর দিন গোটাটাই তো এই স্ক্রিপ্ট নিয়েই। স্ক্রিপ্ট বলতে, মোটামুটি অর্থে, পরপর নিজে নিজে কম্পিউটারের করে যাওয়ার কাজের তালিকা। এই তালিকাটা বানাতে হয় নির্দিষ্ট কিছু নিয়ম মেনে। এর কিছু নিয়ম হল কমান্ড ইন্টারপ্রিটারের, আর কিছু নিয়ম স্ক্রিপ্টিং ভাষাটার। কমান্ড ইন্টারপ্রিটারের কথা এসেছিল এক নম্বর দিনের শেষে — যে আমাদের আদেশ মেশিনের বোঝার মত করে পৌঁছে দেয়। শেল হল কমান্ড ইন্টারপ্রিটার। এই শেলের নিজের কিছু নিয়মকানুন থাকে। আর স্ক্রিপ্ট দিয়ে একটা বিপুল পরিমাণ প্রোগ্রামিং কাজই করে নেওয়া যায়, করা হয়ও। সেই করার স্বার্থে ওই স্ক্রিপ্ট লেখার ভাষার কিছু ব্যাকরণ বানানো থাকে। সেগুলোও মানতে হয় স্ক্রিপ্ট লেখার সময়। স্ক্রিপ্ট লিখে প্রোগ্রামিং কাজ করে কমান্ড ইন্টারপ্রিটার দিয়ে, একে বলে ইন্টারপ্রিটেড ভাষায় কাজ করা। আর সি ইত্যাদি ভাষায় লেখা প্রোগ্রাম বানাতে হয় কম্পাইলার দিয়ে, তাকে বলে কম্পাইলড ভাষায় কাজ করা। এই দুটোর পার্থক্য নিয়ে বড় আলোচনা আছে দশ নম্বর দিনে গিয়ে। এই শেল স্ক্রিপ্টিং দিয়ে কত কত কাজ যে করে নেওয়া হয় বা করে নেওয়া যায় তার কোনো ইয়ত্তা নেই। পরে আমরা আসব এই আলোচনায়।

যা বলছিলাম, ইনিটের প্রাথমিক দায়িত্ব এইরকম কিছু স্ক্রিপ্ট চালানো। ইনিট এই স্ক্রিপ্টগুলোকে চালিয়ে দেখে নেয় ডিস্ক নেটওয়ার্ক কার্ড ডিসপ্লে কার্ড সাউন্ড কার্ড ইত্যাদি ভৌত রসদগুলো রসদগুলো মেশিনের মধ্যে সব ঠিকঠাক আছে কিনা। হার্ডওয়ার ইন্টারাপ্ট, সিস্টেম কল, ইত্যাদি যে ব্যাপারগুলো আমরা উল্লেখ করেছি দুই নম্বর দিনে, সিস্টেম বাসকে ঘিরে সেগুলো সঠিকভাবে কাজ করছে কিনা, এই স্ক্রিপ্টদের দিয়ে সেটাই পরখ করে নেয় ইনিট। বা কনফিগারেশনে যাদের যাদের পাওয়ার কথা তাদের ঠিকঠাক পাচ্ছে কিনা। ডিভাইস ড্রাইভারের যে যে মডিউলগুলো লোড করছে কারনেল তাদের সঙ্গে সিস্টেমের ভৌত ডিভাইসের কনফিগারেশন মিলছে কিনা, ইত্যাদি। এমনকি মডিউলগুলোর সঙ্গে জড়িত কোনো লাইব্রেরি শেষবার বুটের পরে যদি বদলে গিয়ে থাকে সেটাও আপডেট করে নেওয়ার কথা থাকে এই স্ক্রিপ্টে (এই যে কথাগুলো পড়ে যাচ্ছেন এর অনেকটাই আপনার হিব্রু বলে মনে হতে পারে, এমনকি এদের সম্পর্কে পরিচিতিগুলো পাঠমালার গোড়ার দিকে পড়ে আসার পরেও। কারণ এদের এখনো জানেন আপনি, চেনেন না, এখনো এরা আপনার মাথার বাইরে বসবাস করছে। বারবার পেতে পেতে একসময় দেখবেন মাথার ভিতরে ঢুকে গেছে, মিলে যাচ্ছে সব, কিন্তু সেটা ঘটবে এই পাঠমালায় না, আপনার মাথার মধ্যে। সেখানে পাঠমালাটা শেষ, আপনার নিজের যাত্রার শুরু)। শেষবার যদি মেশিন ঠিকভাবে বন্ধ বা শাট-ডাউন না-হয়ে থাকে, তাহলে ডিস্কে লেখা ফাইলব্যবস্থায় কিছু কিছু গোলযোগ চলে আসতে পারে। হঠাৎ কারেন্ট চলে গিয়ে এটা ঘটতে পারে, অন্য কারণেও ঘটে থাকতে পারে। ডিস্কের শরীরে লিপিবদ্ধ ফাইলসিস্টেমে কোনো গোলযোগ আছে কিনা, সেটাও খতিয়ে দেখে নেয় ইনিট তার এই ঘরগেরস্থালি গুছিয়ে নেওয়ার প্রক্রিয়ায়। কখনো কখনো প্রয়োজন অনুযায়ী কোনো শুশ্রূষার কাজ করতে হবে কিনা সেটা যাচাই করে নেওয়াও ইনিটেরই কাজ।

ইনিট এবার জাগিয়ে তোলে কিছু যক্ষ বা যখকে। কুবেরের ধনের মত লিনাক্স সিস্টেমকে অতন্দ্র প্রহরা দিয়ে যাওয়াই কাজ এই ডিমন বা যখগুলোর। কোনো যখ পাহারা দেয় প্রিন্ট স্পুলারকে (স্পুল কাকে বলে মনে করে নিন), এর নাম প্রিন্ট ডিমন (printd)। কোনো যখ পাহারা দেয় মেইলের নড়াচড়াগুলো, কোথা থেকে কোন মেইল কোথায় যাচ্ছে, তার নাম মেইল ডিমন (maild)। কোনো যখ পাহারা দেয় সময়সারনী মোতাবেক করার কাজ গুলোকে, ঠিক সময়ে কাজগুলো শুরু আর শেষ হচ্ছে কিনা, তার নাম ক্রন ডিমন (crond), ইত্যাদি, মনে পড়ছে? এই পাহারার কাজে কোনো যখেরই কখনো কোনো ভুল হয়না, স্বাধিকারপ্রমত্ত হয়না তারা কখনো, তাই তাদের কুবেরের হাতে শাস্তি পেতে হয়না, নির্বাসিত হতে হয়না, কান্তাবিরহগুরুভারে হাহুতাশ করতে হয়না। হায়, তাই কম্পিউটারের যখেদের কোনো মেঘদূত কাব্য লেখা হয়না।

এই ডিমন বা যখ-গুলো অনেকগুলো পরস্পর প্রতিযোগী কাজের বিভিন্ন ধরনের রসদ-চাহিদার উপরও নজর রাখে। কোনো দুটো প্রোগ্রাম একই সঙ্গে একই রসদ দাবি করে কোনো গোলযোগ বাধাচ্ছে কিনা সেটাও নজর রাখতে হয় এই যখদের। এইভাবে যখ বা সদাজাগ্রত ডিমনের আকারে এই ধরনের প্রোগ্রামকে রাখার কারণ এই যে, একসঙ্গে অনেকগুলো প্রোগ্রাম যখন চলছে, তাদের প্রত্যেকটার রসদচাহিদার উপর নজর রাখার জন্যে এক একটা আলাদা আলাদা প্রসেস চালু করা, এবং সেই প্রসেসগুলো আবার যাতে এ অন্যকে ঘেঁটে না-দেয় সেই ব্যবস্থা করার চেয়ে

বোধহয় সহজ হয় এমন একটা প্রোগ্রাম বানানো যা কখনোই থামবেনা। যতক্ষণ কম্পিউটার চলবে সেও চলবে, এবং কনস্টান্ট নজর রেখে যাবে সমস্ত প্রসেসের সদা বদলাতে থাকা সমস্ত চাহিদার উপর। কোনো একটা বিশেষ মেশিন চালু হওয়ার সময় কী কী ডিমন সেখানে চালু হবে এটা মেশিন থেকে মেশিনে আলাদা। ওই মেশিনে কী অপারেটিং সিস্টেম, কী কী প্রোগ্রাম সেখানে চলবে, কী কী কাজ করার আছে সেই মেশিনে, কী কী ধরনের হার্ডওয়ার সেখানে আছে — এই সবকিছুর উপর নির্ভর করে বদলে যায় যখদের তালিকা।

বুটপ্রক্রিয়ায় এবার আসে বিভিন্ন ব্যবহারকারীর জন্যে সিস্টেমকে প্রস্তুত করে তোলার পালা। এই ধাপে এসে গেটি চালু হয়। ইনিটের চালু করা গেটি নজর রাখতে শুরু করে টেলিটার্মিনাল বা কনসোলের উপর। গেটি কিন্তু একাধিক হতে পারে। গ্নু-লিনাক্সে, আগেই বলেছি, একটা প্রোগ্রাম চলা মানে তার মূল প্রোগ্রাম ফাইলটার একটা কপি একবার সক্রিয় হয়ে ওঠা। একটা প্রোগ্রামের একটা কপি একবার চলছে — এর মানে একটা প্রসেস। গেটি প্রসেস সেইরকম একাধিক চালু হতে পারে। ধরুন, মেশিনটা নেটওয়ার্কে লাগানো আছে। এই অবস্থায়, ডায়াল-ইন করে বা অন্য কোনো রকম নেটওয়ার্ক সংযোগ ব্যবহার ভৌগোলিক ভাবে দূরবর্তী কোনো ব্যবহারকারীও লগ-ইন করতে পারে সিস্টেমে। বাইরে থেকে কোনো ব্যবহারকারীর প্রবেশ ঘটছে কিনা কোনো কনসোলে সেটাও নজরে রাখার দায়িত্ব গেটির। তাই, গেটিকে নজর রাখতে হয় সিরিয়াল পোর্টেও। ব্যবহারকারীর প্রবেশ বা লগ-ইন করার গোটা খতিয়ানটা রাখে গেটি। যদি একাধিক টেলিটার্মিনাল মানে কনসোল থাকে তাহলে তাদের প্রত্যেকের জন্যেই এক কপি করে গেটি প্রসেস চালু হয়। গেটিই কনসোলে ফুটিয়ে তোলে লগিন প্রম্পট। এখানে আমরা ম্যানড্রেক ৯.১-এর লগ-ইন প্রম্পট দেখালাম। সমস্ত ডিমনগুলো এবং টেলিটার্মিনাল পিছু একটা করে গেটি প্রসেস চালু হওয়ার পর শেষ হয় ইনিট এর ১ নম্বর রানলেভেলের কাজ। রানলেভেল ২ বা ৩ বা ৫-এ যা যা প্রক্রিয়া চালু হয়, খেয়াল করে দেখুন একবার, তাদের মধ্যে কিন্তু রানলেভেল ১-এর সবগুলো প্রক্রিয়াই আছে। এবং আরো কিছু আছে তাদের প্রত্যেকটাতেই যা যা ১ নম্বর রানলেভেলে নেই। তাই রানলেভেল ১-এর প্রক্রিয়াগুলো চালু হবেই, বুট হওয়ার একটা স্তর অব্দি রানলেভেল ১-এর মত করে গোটাটা চালু থাকে। তারপর শুরু হয় '/etc' ডিরেক্টরিতে 'inittab' ফাইলে দেখানো ডিফল্ট রানলেভেলের জন্য নির্দিষ্ট তালিকা মেনে বাড়তি প্রক্রিয়া বা যখ বা পরিষেবাগুলো চালু হওয়ার কাজ। প্রতিটি রানলেভেলের সঙ্গে সঙ্গতি রেখে এই রকম একটা করে তালিকা তৈরি থাকে সিস্টেমে। আবার, নিজের ইচ্ছে হলে বিশেষ কোনো প্রসেসকে আপনি চালু বা বন্ধ করতে পারেন সেই তালিকা থেকে। ধরুন সেই ডিফল্ট রানলেভেল যদি ২ হয়, মানে 'inittab' ফাইলে লাইনটা যদি হয়, '`id:2:initdefault:`', তার মানে, এই ২ রানলেভেলের মত একাধিক ব্যবহারকারীর সিস্টেম চালু করতে হবে। অর্থাৎ, রানলেভেল ১-এর একক ব্যবহারকারী মানে মূল বা রুট ইউজার আছে এখানেও। তার সঙ্গে চালু হবে বাড়তি ব্যবহারকারীকে ঢুকতে এবং কাজ করতে দেওয়ার প্রক্রিয়া। সেগুলো চালু করে এবার রানলেভেল ২ শুরু হবে। এবার রুট ইউজার ছাড়াও অন্য একজন সাধারণ ইউজার এবার সিস্টেমে লগ-ইন করতে এবং কাজ শুরু করতে পারবে।

এখানেও কিন্তু গোটা বুটপ্রক্রিয়াটা না শেষ হতে পারে। ধরুন আপনার মেশিনে আপনি '/etc' ডিরেক্টরিতে 'inittab' ফাইলে ডিফল্ট রানলেভেল রেখেছেন ৩, মানে 'inittab' ফাইলে লাইনটা রয়েছে '`id:3:initdefault:`'। রানলেভেল ৩ মানে রানলেভেল ১ এর প্রক্রিয়াগুলো তো আছেই, তার সঙ্গে একাধিক ব্যবহারকারীর জন্যে প্রয়োজনীয় প্রক্রিয়াগুলো এবং নেটওয়ার্কের জন্যে দরকারি প্রক্রিয়াগুলোও চালু হবে। রানলেভেল ৩-এ বহু ইউজার আর নেটওয়ার্ক দুটোই আছে। তার মানে, রানলেভেল ১-এর প্রাথমিক প্রক্রিয়াগুলোর উপর এখন চালু হবে মাল্টিইউজার আর নেটওয়ার্ক সংক্রান্ত ডিমনগুলো অন্য প্রাসঙ্গিক প্রসেসরা। তবে শুরু হবে রানলেভেল ৩ — বুটপ্রক্রিয়া শেষ হবে। যদি কোনো কম্পিউটারে ব্যবস্থা এমনই করা থাকে যে সেখানে মাত্র একজন ব্যবহারকারী এবং নেটওয়ার্ক ইত্যাদি সেখানে নেই, (সেই গ্নু-লিনাক্স কিন্তু গ্নু-লিনাক্স নয়, একটা কবন্ধ মাত্র) সেখানে এই রানলেভেল ৩-এর ক্রিয়াগুলো আর ঘটবেনা। তবে এই কবন্ধ গ্নু-লিনাক্স-ও মাঝে মাঝে প্রয়োজন পড়ে। সিস্টেমে কোনো কেলো ঘটলে, সেটা সামলানোর জন্যে এরকম একটা মুন্ডুকাটা সিস্টেমে ঢুকে ত্রুটিগুলো সংশোধন করার দরকার পড়ে।

```
Mandrake Linux release 9.1 (Cooker) for i586
Kernel 2.4.19-24mdk on an i686 / tty1
localhost login:
```

৮।। তুমি কে গা? — লগ-ইন

থেবাই-এর স্পিংক্স-এর মত ঘাপলা ঘাপলা ধাঁধা জিগেশ করেনা বটে, কিন্তু অয়দিপাউসের সেই দানো যেমন অয়দিপাউসের কথা অয়দিপাউসের চেয়েও বেশি জানত, আমাদের এই সিস্টেমও তাই। আমাদের প্রত্যেকের কাজ, কাজের জায়গা, কাজের স্বাধীনতা, অধিকার, অনুমতি — এসব সে আমাদের নিজেদের চেয়েই অনেক বেশি খেয়াল রাখে। নথী বানিয়ে রাখে আমাদের প্রত্যেকের জন্যে। নথী মেলানোর জন্যে শুধু, প্রথমেই জানতে চায়, তুমি কে গা?

```
login:
```

প্রম্পটের এই চেহারাটা প্রমাণ করে, এই কনসোলটা এই মুহূর্তে ফাঁকা আছে। কোনো ইউজার এখন ঢুকতেই পারে সেখানে। কোলনের পর দপ দপ করতে থাকা কার্সর আমাদের জিগেশ করছে — নিজের নাম বসাতে হবে এবার। যে কোনো নাম নয়, যে নামে সিস্টেম আমাকে চেনে। সিস্টেমের মধ্যে যাবতীয় ব্যবহারকারীর প্রয়োজনীয় সমস্ত খুঁটিনাটির তালিকা বানানো থাকে। কোনো সিস্টেম ব্যবহার করার আগে সেই তালিকায় নিজের নাম তুলে দিতে হয়। ইউজারঅ্যাড (`useradd`) কমান্ড দিয়ে সিস্টেমের সেই তালিকায় নাম তোলার সময় যে নাম তাকে দেওয়া হয়েছিল, সেই নামটা এবার টাইপ করে দিতে হবে লগ-ইন প্রম্পটে। মেশিনে গ্নু-লিনাক্স কোনো ডিস্ট্রিবিউশন ইনস্টল করার মধ্যে অনেক সময় আমরা সরাসরি এই ‘useradd’ কমান্ডটা ব্যবহার করিনা। ইউজার যোগ করার উইনডোর ছবির মধ্যে নামটুকু টাইপ করে দিয়ে ইঁদুর ক্লিক করে নতুন ইউজারের নাম যোগ করি সিস্টেমে। কিন্তু গুই এর এই ইঁদুর-আর-ছবির প্রোগ্রামটা আসলে অন্তরালে এই ইউজারঅ্যাডকেই ব্যবহার করে।

যদি মেইনফ্রেম সিস্টেম হয়, যেখানে ব্যবহারকারীর সংখ্যা অনেক, সেখানে নিজেকে করতেই হয়না কাজটা। সিস্টেম অ্যাডমিনস্ট্রেটর নিজেই আমার প্রাথমিকতম পরিচয়টা তালিকায় ঢুকিয়ে দিয়েছে, সেটাও ওই ইউজারঅ্যাড ব্যবহার করে। আর আপনার আমার বা অন্য অনেকের নিজের টেবিলে রাখা নিজের ব্যক্তিগত পিসিতে, ব্যক্তিগত সিস্টেমে ব্যবস্থাটা নিজেই করতে হয়েছে, ইনস্টলেশনের সময় বা পরে। ‘useradd’ কমান্ডটাও একটা সিস্টেম প্রোগ্রাম, তাই রুট ছাড়া আর কেউ ব্যবহার করতে পারেনা। প্রথমে ‘`su`’ কমান্ড দিয়ে রুট হয়ে, রুট বা অ্যাডমিনস্ট্রেটর-এর পাসওয়ার্ড দিয়ে, তবেই আর একজন ইউজার যোগ করা যায়। পরে আমরা আরো ভালো করে বুঝব, একটা গ্নু-লিনাক্স সিস্টেমের সুপারইজার বা রুট নামের ব্যবহারকারী হল চূড়ান্তভাবে একচ্ছত্র, কোনো কিছু করাতেই তার কোনো বাধা নেই, যে কোনো ফাইল যে কোনো সময়ে সে যে কোনো ভাবে বদলে নিতে পারে।

কিন্তু ‘বদলে নিতে পারা’ মানে তো ‘বদলে ফেলতে পারা’-ও বটে। তাই, এমনকি নিজের ব্যক্তিগত সিস্টেমেও কখনো রুট হয়ে কাজ করতে নেই। ভুল প্রত্যেকের হয়। রুট হয়ে সিস্টেম ব্যবহার করার সময়, কোনো মুহূর্তের ভুলে, গোটা সিস্টেমকে অসংশোধনযোগ্য রকমে ঘেঁটে ফেলতে পারি আমরা, যে কোনো মুহূর্তে। এমনকি উড়িয়েও দিতে পারি গোটা সিস্টেমটা। এখানে বলে রাখি, যে কোনো ইউনিক্স-এই ‘রুট’ শব্দটা এই দুটো অর্থেই ব্যবহার হয়, এক, সুপারইউজার বা সর্বশক্তিমান চূড়ান্ত ব্যবহারকারী, আর দুই, রুট ডিরেক্টরি, যার কথা আমরা আগেই বলেছি লিলোর আলোচনায়, যে ডিরেক্টরির মধ্যে আছে আর সমস্ত ডিরেক্টরি। যাই হোক, যা বলছিলাম, তাই নিরাপদ উপায় হল এমনিতে নিজের জন্য সাধারণ একজন ব্যবহারকারী বা ইউজার বানিয়ে কাজ করা, এবং কোনো সিস্টেম প্রোগ্রাম ব্যবহার করার সময়টুকু ‘su’ করে সুপারইউজার বা রুট হয়ে নেওয়া। ‘useradd’ কমান্ড দিয়ে নিজের জন্যে একটা ইউজার যোগ করে নেওয়ার পরে, তার সিস্টেমে লগ-ইন করার সঙ্কেত বা পাসওয়ার্ডটাও বানিয়ে নিতে হয়। তার কমান্ডটা হল ‘`passwd`’। পরেও কখনো নিজের পাসওয়ার্ড বদলাতে চাইলে এই ‘passwd’ ব্যবহার করেই সেটা করা যায়। এই ‘su’, ‘useradd’, বা ‘passwd’ কমান্ডগুলো সম্পর্কে আর একটু জানতে চান? ‘man’ ব্যবহার করুন।

চার নম্বর দিনের থেকে মনে করুন, এক ধরনের বহু টার্মিনালের বহু ব্যবহারকারীর মেশিনে ব্যবহারযোগ্য ওএস হিশেবে বেড়ে উঠেছিল ইউনিক্স। ওরকম সিস্টেমে কম্পিউটার শেখার সুযোগ হয় খুব কম জনেরই। কিছু রিসার্চ ইনস্টিটিউট বা খুব কম দু-একটা অফিস। তারা অনেক ব্যাপক এবং গভীর ভাবে শেখে সিস্টেমকে। এরা এই পাঠমালার টার্গেট পাঠক নয়, এটা আমার মত গোলা লোকেদের জন্যে, যাদের কম্পিউটার শেখা এবং ব্যবহারটা এক ইউজারের এক কনসোলের ব্যক্তিগত স্ট্যান্ড-অ্যালোন পিসিতে, যারা সচরাচর অন্য মেশিনের সঙ্গে নেটওয়ার্কে সংযুক্তও থাকেনা। কখনো কখনো অনলাইন হয়, ইন্টারনেটে পৌঁছতে, কিন্তু নেটওয়ার্কটা সেখানে সর্বব্যাপী নয়।

এরকম সিস্টেমে সচরাচর একটা করে কনসোল থাকে। সেই একটা কনসোলকেও গ্নু-লিনাক্স একাধিক ভুতুড়ে পট মানে ভার্চুয়াল কনসোল করে তুলতে পারে, একের বেশি ভার্চুয়াল ব্যবহারকারী যেখানে লগ-ইন করতে পারে। যেন সেটা বহু কনসোলের বহু ইউজারের একটা মেশিন। কী করে এটা করা যায় সেই কথায় পরে আসছি। আগে ব্যাপারটা বুঝুন। এদের ভৌতিক ব্যবহারকারী বা ভার্চুয়াল ইউজার বলছি এই অর্থে যে একই কিবোর্ড থেকে কমান্ড দিয়ে একই স্ক্রিনে নিজেদের আদেশের ফলাফল দেখে কাজ চলছে সেখানে। বাস্তবে একটা লোকই সেটা করছে। কিন্তু সিস্টেম তাদের দেখছে একাধিক ব্যবহারকারী হিশেবেই। যেমন একটা বাস্তব বহু-টার্মিনাল বহু-ইউজার মেশিনে হত। ধরুন, আমার সিস্টেমে চারজন ব্যবহারকারী, ‘atithi’, ‘dd’, ‘manu’ এবং ‘piu’। আর যে কোনো সিস্টেমেই একজন ‘root’ মানে সুপারইউজার বা রুট তো থাকবেই। এবার পাঁচটা ভৌতিক টার্মিনালে পাঁচজন ব্যবহারকারী হিশেবে লগ-ইন করে পাঁচটা কাজ ধরুন চালু করলাম। এবার এর, মধ্যে কোনো একটা ভৌতিক টার্মিনাল থেকে কমান্ড দিলাম, ‘w’। ‘w’ কমান্ডটা একটা তালিকা তৈরি করে সিস্টেমে কে কে লগ-ইন করে রয়েছে এবং তারা কী কী করছে। দেখুন, প্রথম লাইনটা দেখাচ্ছে কটায় কমান্ডটা চালানো হয়েছে, কত সময় ধরে মেশিন চলছে, কত জন ইউজার ব্যবহার করছে, এবং শেষ কিছু সময়ে তারা মোট কতটা চাপ দিয়েছে সিস্টেমের উপর। দ্বিতীয় লাইনটা দেখুন, সেটা এক সারি হেডিং। প্রথম হেডিং ‘USER’ মানে ইউজারের নাম, দ্বিতীয়টা হল ‘TTY’ মানে কোন টার্মিনালে সে লগ-ইন করে রয়েছে, তৃতীয়টা হল ‘LOGIN@’ মানে কটার সময় সে লগ-ইন করেছে, চতুর্থটা ‘IDLE’ মানে কত সময় হল সে চুপচাপ বসে আছে, পঞ্চম আর ষষ্ঠটা হল সে সিপিইউকে কী রকম ব্যবহার করছে, আর শেষটা হল ‘WHAT’, মানে সে কী করছে। শুধু একটা জিনিষ খেয়াল করুন, এখানে ‘root’ চালিয়েছে ‘w’ কমান্ডটা, সে নিজে আছে চার নম্বর টার্মিনালে, আর সেই কমান্ডেরই ফলাফল এখানে লেখা, তাই একমাত্র তারই অলস সময় শূন্য, কারণ সেই মুহূর্তেই তো দেওয়া হয়েছে কমান্ডটা। এখুনি সবগুলো উপাদান বোঝার চেষ্টা করবেন না, প্লিজ। শুধু যে কথাটা বলছিলাম, দেখুন, একটা টার্মিনালের একা-একলা একটা ব্যক্তিগত পিসিকেও গ্নু-লিনাক্স সিস্টেম কিন্তু একটা পূর্ণাঙ্গ সিস্টেম হিশেবে নাড়াচাড়া করছে। যেখানে অনেক ব্যবহারকারী আছে, এমনকি নেটওয়ার্কও আছে, রিমোট লগ-ইন করা যায়। আপনার গ্নু-লিনাক্স সিস্টেমে ‘man w’ কমান্ড দিয়ে ‘w’ কমান্ডের ম্যানুয়াল পড়ে দেখুন, ‘LOGIN@’ মানে তৃতীয় স্তম্ভটা সেটাও দেখাত যদি কেউ রিমোট লগ-ইন করা থাকত এখানে। রিমোট লগ-ইন বলতে নেটওয়ার্কে সংযুক্ত অন্য কোনো মেশিন থেকে যদি কেউ এই সিস্টেমে লগ-ইন করা থাকত সেই খুঁটিনাটিটাও ধরে দিত ‘w’।

```
18:01:52 up 8 min,  5 users,  load average: 0.00, 0.03, 0.02
USER     TTY      LOGIN@   IDLE   JCPU   PCPU  WHAT
dd       tty1     17:54    7:16   0.16s  0.08s links .
manu     tty2     17:54    1:36   0.79s  0.01s login
atithi   tty3     17:56    5:36   0.02s  0.02s -bash
root     tty4     17:56    0.00s  0.21s  0.00s w
piu      tty5     18:01   32.00s  0.02s  0.02s -bash
```

এবার ভৌতিক পট বা ভার্চুয়াল টার্মিনাল খুলে ভার্চুয়াল ইউজারের ভার্চুয়াল লগ-ইনের তরকিবটা বাতলে দেওয়া যাক। ‘<Ctrl>+<Alt>+<F1>’ থেকে ‘<Ctrl>+<Alt>+<F6>’। মানে একইসঙ্গে কনট্রোল আর অল্ট আর একদম উপরের সারির একটা এফ-সুইচ, এক থেকে ছয়, পরপর আমাদের ভার্চুয়াল কনসোল ১ থেকে ৬ অব্দি নিয়ে যায় বা ফেরত আনে। কিবোর্ডে সবচেয়ে নিচুতলার বামপন্থীতম চাবিটাই হল কন্ট্রোল (<Ctrl>)। কমিউনিস্ট স্ট্রেটজ্যাকেটিং নিয়ে এটা কিবোর্ড-দেবতার কোনো গোপন মন্তব্য কিনা তা ফুকোদা বেঁচে থাকলে বলতে পারতেন। আর এই কনট্রোল চাবিটার থেকে ঠিক টু-স্টেপ-ব্যাক, দুই পা ডানদিকে পিছনে, দক্ষিণপন্থী বিচ্যুতিতে ভোগা সুইচটাই অল্ট (<Alt>)। এফ-ওয়ান (<F1>) থেকে এফ-সিক্স (<F6>) পাবেন একদম উপরের সারিটায় বাঁদিকে, এসকেপ সুইচটার পরেই। এই হল ছ-টা ভৌতিক পটের হিশেব। আর সাত নম্বর ভার্চুয়াল কনসোলটা তোলা থাকে গুই বা এক্স-উইনডোজের জন্যে।

এই ছয়পিস ভৌতিক পটের প্রত্যেকটাতেই আলাদা আলাদা করে লগ-ইন করতে পারে আলাদা আলাদা বা একই ব্যবহারকারী এবং কাজ করে চলতে পারে। যদি আপনি আমারই মত আইডেন্টিটি ক্রাইসিসে ভোগেন, নিজের নাম দেখতে খুব ভালোবাসেন, বারবার করে, তাহলে ছ-খানা কনসোলেই আলাদা আলাদা করে লগ ইন করুন। এবং

তারপর কমান্ড দিন ‘`who`’। একটু আগের ‘w’ কমান্ডটার মতই ‘who’ কমান্ডটা ফুটিয়ে তোলে কে কে লগ-ইন করে রয়েছে সিস্টেমে, শুধু তাদের অতটা ডিটেইলস আর দেখায়না ‘w’ কমান্ডের মত। ‘who’ কমান্ড দেওয়ার পর দেখবেন ছবার আপনার নাম ফুটিয়ে তুলেছে সিস্টেম, পরপর ছ লাইনে। নিজেকে একই সঙ্গে ছজন বলে মনে করতে চাইলে, সেই কমান্ড-ও তো আগেই বলেছি, একটা কমান্ড সমাহার, ‘who’ আর ‘wc’ এই দুটো কমান্ডকে একসঙ্গে মিলিয়ে কাজ করে, ‘`who|wc -l`’। এটা পরে আমরা পাইপিং-এর আলোচনায় দেখব, ইউনিক্স তথা গ্নু-লিনাক্স সিস্টেমের ইতিহাসে সবচেয়ে বিখ্যাত কমান্ডগুলোর একটা হল এই ‘who|wc –l’। সিস্টেমে কতজন লগ-ইন করা আছে সেটা গুণে দেয়। এটা এর কারণ, আগেই যেমন বলেছি, প্রত্যেকটা ভৌতিক পট বা ভারচুয়াল কনসোলেই একখানা করে গেটি প্রসেস কাজ করছে। যাদের প্রত্যেকটাতেই এক এক জন আলাদা আলাদা আপনি লগ-ইন করেছেন, ছ বার, ছ-টা আলাদা জায়গায়। আপনার একটুও কৌতূহল হচ্ছেনা কমান্ডদুটো সম্পর্কে আর একটু জানার?

আবার ফেরত আসা যাক লগ-ইনের কথায়। প্রবেশ বা লগ-ইনের সময় সেই নামটাই দিতে হবে, যে নামে ইউজার বানানো এবং যোগ করা হয়েছিল। লগ-ইনের সময় যদি অন্য নাম দিই আমরা, ছোটদিদা মারা যাওয়ার পরে যে নামে আর কেউ ডাকেনি ততটা না-হলেও অন্তত যে নামে আপনি আপনাকে সিস্টেমে কোনোদিন ডাকেননি, সরি, এই স্পিংক্স তাকে চেনেনা, অতটা অর্ন্তযামী নয়। হতে পারে গ্নু-লিনাক্স, তাই বলে আশা করার তো একটা সীমা থাকবে। কিন্তু, মালটা হেভি ধূর্ত, অন্তর্যামী যে নয় সেটা বুঝতে দেবেনা আপনাকে। বা হয়ত একটু ইগো ক্রাইসিসে ভোগে, নিজের না-জানাটা স্বীকার করতে চায়না। যে নামই দিইনা কেন, এখন কিছু বলবেনা, পরের লাইনে সোনামুখ করে জানতে চাইবে

```
password:
```

আবার সেই দপদপ করতে থাকা নির্বাক প্রশ্ন। এর উত্তরে নিজের সঙ্কেত বা পাসওয়ার্ডটা দিতে হবে। যে পাসওয়ার্ড তাকে জমা দেওয়া হয়েছিল ‘passwd’ কমান্ড দিয়ে। মনে পড়ছে? এবার মজাটা এইখানে, ইউজারনেম আর পাসওয়ার্ড এই দুটো জিনিষের, যেকোনো একটা ভুল হলেই কম্পিউটার এবার জানাবে লগ-ইন ঠিক হয়নি। এবং নতুন করে আবার লগ-ইন করার জন্যে ব্যবহারকারীর নাম বা ইউজারনেম চাইবে। কিন্তু কখনোই আপনি আলাদা করে জানতে পারবেন না, দুটোর ভিতর ঠিক কোনটা ভুল হয়েছে।

```
login incorrect
login:
```

অর্থাৎ, আমরা একবারের তরেও জানতে পারলাম না, গন্ডগোলটা ঠিক কোথায় হয়েছিল ইউজারনেমে না পাসওয়ার্ডে। এটা ইউনিক্স সিস্টেমগুলোর একটা বাড়তি নিরাপত্তা ব্যবস্থা। বাইরে থেকে কেউ যদি ঢুকতে চায় সে যাতে একটুও বাড়তি তথ্য না পায়। আর লগ-ইন নেম এবং পাসওয়ার্ড দুটোই যদি ঠিক হয় এবার তিনি জানাবেন

```
Last login: Thu Oct 23 23:05:02 on tty1
```

দার্জিলিং মেলের ফার্স্ট ক্লাস কামরায় চৌকো জানলার সামনে ব্লাউজে পেন গুঁজে মাথা ঝুঁকিয়ে গালে টোল ফেলেনা সত্যিই, কিন্তু দেখেছেন, সিস্টেম আপনাকে কতটা ‘মনে রেখে দেয়’, আপনি নায়ক উত্তমকুমার না-হওয়া সত্ত্বেও। ঘ্যাচাঘ্যাঁচ জানিয়ে দিচ্ছে শেষ আপনি কখন সিস্টেমে ঢুকেছিলেন। এর সঙ্গে এক এক সিস্টেম এক এক সেট একস্ট্রা তথ্যও দেয়। কতটা তথ্য দেবে সেটাও নিয়ন্ত্রণ করা যায়। এই নিজের ইচ্ছে বা প্রয়োজন অনুযায়ী সিস্টেমের যে কোনো জায়গা বদলে নেওয়ার উপায়টা হল কনফিগারেশন ফাইল বদলে বদলে। গ্নু-লিনাক্স ব্যবস্থায় এটা প্রায় সর্বব্যাপী। পরে আমরা শিখব এটা, পাঠমালার একদম শেষ দিকে গিয়ে।

লগ-ইন করা মাত্র, কোনো কাজে হাত দেওয়ারও আগে, ইতিমধ্যেই, আপনার জন্যে একটা প্রোগ্রাম কিন্তু চালু হয়ে গেছে। সেটার নাম শেল। সচরাচর গড় গ্নু-লিনাক্স ব্যবহারকারীর শেলটার নাম ব্যাশ, কমান্ড ইন্টারপ্রিটার। আপনার দেওয়া কমান্ডকে যে ইন্টারপ্রিট করে পাঠায় সিস্টেমের কাছে। দিন নম্বর এক-এ আমরা অল্প আলোচনা করেছি এই নিয়ে। পরে আবার আসব। যে জায়গাটায় আপনি এখন আছেন, মানে কিবোর্ডে টাইপ করলেই সেই অক্ষরগুলো যেখানে ফুটে উঠবে, সেই জায়গাটার নামই কমান্ড প্রম্পট। একটা কথা খেয়াল রাখবেন, আপনি যে কমান্ডটা দিচ্ছেন, সেটা কিন্তু সরাসরি এখানে ফুটে উঠছে না। আপনি এন্টার মারা মাত্র আপনার কমান্ডটা পৌঁছে যাচ্ছে

সিস্টেমের কাছে। সেই কমান্ডের একটা প্রতিরূপ সে পাঠিয়ে দিচ্ছে স্ক্রিনে। আপনার দেখার জন্যে, শেষ কী কমান্ড গেছে। আর অন্য একটা প্রতিরূপ সে পাঠিয়ে দিচ্ছে ব্যাশ বা অন্য কোনো কমান্ড ইন্টারপ্রিটারের কাছে। কমান্ডের এই দুটো প্রতিরূপ একই সঙ্গে ব্যবহার করার টেকনিকাল নাম 'ডাবল ডুপ্লেক্স' (double-duplex)। এর আবার নানা বিশেষ তাৎপর্য আছে সিস্টেমে। জায়গাটা বেশ মজার। কিন্তু আমাদের এই পাঠমালায় আমরা অতটা দূর অব্দি যাবনা। নামটা দিয়ে রাখলাম যাতে কোথাও পেলে অপরিচিত না-লাগে। যাইহোক, যেখানে আপনার কমান্ডটা ফুটে উঠছে, সেই কমান্ড প্রম্পট-টা নানা চেহারা নিতে পারে।

`$_` বা `#_` বা `user@linux$_`

এদের কোনটা শেষ অব্দি ফুটে উঠবে সেটা নির্ভর করে কী ডিস্ট্রিবিউশন আপনি ব্যবহার করছেন তার উপর, মানে সেটা রেডহ্যাট না ম্যানড্রেক না ডেবিয়ান না সুজে না স্ল্যাকওয়ার ইত্যাদি, আর সেই মুহূর্তে আপনি সাধারণ ব্যবহারকারী না রুট, তার উপরেও। এই প্রম্পটের চেহারাটাও বদলে নেওয়া যায় সহজেই। এই খুঁটিনাটিগুলো এবং প্রম্পটের চেহারা থেকে সিস্টেমের যে কোনো কনফিগারেশনই কী ভাবে বদলাব, তার আলোচনা আসতে এখনো অনেকটা দেরি আছে। এইবার, এই সমস্ত ধাঁধার ঠিকঠাক উত্তর দেওয়ার পরে গ্নু-লিনাক্স সিস্টেমের স্পিংক্স আপনাকে ঢুকতে দিল। আপনি লগ-ইন করলেন, এবার আপনার কাজ করার জন্যে সে ফুটিয়ে তুলল কমান্ড প্রম্পট। এখন আপনি প্রম্পটে আছেন, আপনি কমান্ড দিয়ে যেই এন্টার মারবেন কম্পিউটার তার কাজ শুরু করবে। লগ-ইন শেষ হল।

নিচে আমার মেশিনে সুজে ৮.২-এ একবার লগ-ইনের গোটা লগ-ইন বানীসহ লগ-ইন প্রম্পট-টা তুলে দিলাম। ওই বানীটা আসলে ফরচুন কুকি, ওয়েফার মুড়ে দেওয়া কাগজে লেখা ভবিষ্যত বা অন্য কোনো বাণীর মত — গ্নু-লিনাক্স সিস্টেমের অনবদ্যতম চ্যাংড়ামোগুলোর একটা। এমন ব্যবস্থা করা আছে যে প্রতিবার আমি লগ-ইন করলেই একটি করে ফরচুন-বানী আমায় শোনায়, লগ-ইন মেসেজ সহ। কী করে ফরচুন বা অন্যান্য বাণী শোনানো শুরু বা বন্ধ করা যায়, সেগুলো আমরা পাব পাঠমালার শেষ মানে দশ নম্বর দিনে। এবার এই '`dd@linux:~>`' লেখাটার পর জ্বলতে নিভতে থাকা কার্সরটাই আমার কমান্ড প্রম্পট, যেখানে আমি কমান্ড টাইপ করে এন্টার মারলে আমার মেশিন কাজ করবে।

```
Welcome to SuSE Linux 8.2 (i586) - Kernel 2.4.20-4GB-athlon (tty1).
linux login: dd
Password:
Last login: Thu Oct 23 08:55:42 on tty2
Have a lot of fun...
Anthony's Law of Force:
        Don't force it; get a larger hammer.
dd@linux:~>
```

৯।। লগ-ইনকে একটু খতিয়ে দেখা — পাসওয়ার্ড

সিস্টেমে আমি লগ-ইন করছি মানে গেটির কাছে নিজের পরিচয় জানাচ্ছি। গেটি তখন লগ-ইন (login) নামের প্রোগ্রামটা চালাচ্ছে। লগ-ইন প্রোগ্রামটা সিস্টেমের মধ্যে রাখা তালিকা থেকে আমার নাম আর পাসওয়ার্ড নিয়ে মিলিয়ে দেখছে, আমি আদৌ কোনো ইউজার কিনা, লগ-ইন করার এবং সিস্টেম ব্যবহার করার এক্তিয়ার আমার আছে কিনা। আলাদা করে নয়, দুটোকে একই সঙ্গে মেলাচ্ছে, এবং যদি এর একটাও না-মেলে, লগ-ইন করতে দিচ্ছেনা। আর যদি দুটোই মেলে, মানে, আমার এক্তিয়ার থাকে সিস্টেম ব্যবহার করার, আমাকে প্রবেশ অধিকার দিচ্ছে। এবং আমাকে প্রবেশ করানোর সঙ্গে সঙ্গে তাকে সেরে ফেলতে হচ্ছে একটু ঘরগোছানোর কাজ। মানে, আমার লগ-ইন সংক্রান্ত তথ্য সিস্টেমের 'মনে রেখে দেওয়া'। আর আমার জন্যে একপিস শেল চালু করে দেওয়া। এটুকু করেই সিন থেকে সরে যাচ্ছে লগ-ইন। ঠিক ওই বায়োস আর লিলোর মতই, গেটি আর লগ-ইন একই প্রোগ্রাম হতেই পারত। না-হয়ে তারা আলাদা প্রোগ্রাম হল ওএস বিবর্তনের কিছু ঐতিহাসিক কারণে।

কিন্তু আমার হাতে ওই একখানা খোসা মানে শেল ধরিয়ে দেওয়ার আগে সিস্টেমকে আরো কিছু কাজ করে নিতে হয়। যাতে সে বুঝে নিতে পারে, কোন কোন ফাইলে বা ডিরেক্টরিতে কাজ করার অনুমতি আমার আছে, এবং পরে, কাজ করে চলাকালীন আমি যখনি কোনো ফাইল বা ডিরেক্টরিতে কাজ করতে চাইব, তখনি মিলিয়ে নিতে পারে। এটা কোনো গ্নু-লিনাক্স তথা ইউনিক্স সিস্টেমে সবচেয়ে গুরুত্বপূর্ণ ব্যাপারগুলোর একটা। তবে পুরোনো ইউনিক্সের সঙ্গে গ্নু-লিনাক্সের এই নিরিখে কিছু পার্থক্যও আছে। গ্নু-লিনাক্স গোটা নিরাপত্তা ব্যবস্থাটাই দাঁড়িয়ে আছে এর উপর। আমি নিজেকে চেনাচ্ছি আমার নাম আর পাসওয়ার্ড দিয়ে। এই পাসওয়ার্ড আর নামকে সিস্টেম মিলিয়ে নিচ্ছে '/etc' ডিরেক্টরিতে 'passwd' নামের একটা ফাইলের সঙ্গে। এই ফাইলটা, '/etc/passwd', ঠিক ওই '/etc/lilo.conf' ফাইলের মতই, একটা টেক্সট ফাইল, যার মধ্যে পরপর কয়েক লাইনে প্রতিটি ব্যবহারকারীর বিষয়ে প্রয়োজনীয় তথ্যগুলো রাখা। এখানে এদের দুজনেরই নামের আগে /etc অংশটা জুড়ে দেওয়ার মানে এই যে একটা সিস্টেমে এই দুটো ফাইলকেই পাওয়া যায় রুট ডিরেক্টরি বা ' / '-এর মধ্যে 'etc' নামের ডিরেক্টরিতে। এর আগে অব্দি যাকে লিখছিলাম 'অমুক' ডিরেক্টরির 'তমুক' ফাইল, তাকে বোঝানো যেত 'অমুক/তমুক' ফাইল লিখে। এখানে আমরা লিখেছি '/অমুক/তমুক' ফাইল। এতে বাড়তি এইটুকু বলা আছে যে, 'অমুক' ডিরেক্টরিটা আছে রুট ডিরেক্টরিতে, গ্নু-লিনাক্সে যার নাম '/'। পরে অনেক ভালো করে বুঝতে হবে এসব। এখন একটু সেই '/etc/passwd' ফাইলটা থেকে কয়েক লাইন পড়া যাক।

এটা আমার মেশিনের '/etc/passwd' ফাইল, প্রথম লাইনে আর শেষ চার লাইনে 'atithi', 'dd', 'manu', আর 'piu' ছাড়া আর সবগুলো লাইনেই ব্যবহারকারী হিশেবে যাদের উল্লেখ তাদের প্রত্যেকেই সিস্টেমের বানানো — নিজের কাজ করার স্বার্থে সিস্টেম এই প্রোগ্রামগুলোকে ইউজার বানায়, কারণ, স্বাভাবিক ইউজারদের মত এরাও বিভিন্ন রসদকে ব্যবহার করে, কাজে লাগায়। একদম প্রথম লাইনেই রয়েছে সুপারইউজার বা রুট।

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/bin/bash
daemon:x:2:2:Daemon:/sbin:/bin/bash
lp:x:4:7:Printing daemon:/var/spool/lpd:/bin/bash
mail:x:8:12:Mailer daemon:/var/spool/clientmqueue:/bin/false
news:x:9:13:News system:/etc/news:/bin/bash
uucp:x:10:14:Unix-to-Unix CoPy system:/etc/uucp:/bin/bash
games:x:12:100:Games account:/var/games:/bin/bash
man:x:13:62:Manual pages viewer:/var/cache/man:/bin/bash
at:x:25:25:Batch jobs daemon:/var/spool/atjobs:/bin/bash
wwwrun:x:30:65534:WWW daemon apache:/var/lib/wwwrun:/bin/bash
ftp:x:40:49:FTP account:/srv/ftp:/bin/bash
gdm:x:50:15:Gnome Display Manager daemon:/var/lib/gdm:/bin/bash
postfix:x:51:51:Postfix Daemon:/var/spool/postfix:/bin/false
gnump3d:x:63:65534:GNUMP3 daemon:/var/lib/nobody:/bin/false
sshd:x:71:65:SSH daemon:/var/lib/sshd:/bin/false
ntp:x:74:65534:NTP daemon:/var/lib/ntp:/bin/false
vdr:x:100:33:Video Disk Recorder:/var/spool/video:/bin/false
nobody:x:65534:65533:nobody:/var/lib/nobody:/bin/bash
dd:x:500:100:dipankar das:/home/dd:/bin/bash
manu:x:501:100:Sriparna Das:/home/manu:/bin/bash
atithi:x:502:100: atithi:/home/ atithi:/bin/bash
piu:x:503:100:Smita Bhadra:/home/piu:/bin/bash
```

এই '/etc/passwd' ফাইলের প্রত্যেকটা লাইনেই, লক্ষ্য করুন, একটা কাঠামো আছে। মানে একই ধরনের বিষয় প্রতিটা লাইনেই এসেছে। এরসবগুলো খুঁটিনাটিতে আমরা এখন যাবনা, পরে আসবে। শুধু দেখুন প্রত্যেকটা লাইনেই শুরুতে ব্যবহারকারীর ইউজারনেম, তারপর একটা 'x' , যেটা আসলে তার পাসওয়ার্ডের সূচক। কোনো কোনো সিস্টেমে, যদি নিরাপত্তা নিয়ে ততটা মাথা না-ঘামানো হয়, পাসওয়ার্ডগুলো এখানেই থাকতে পারে, কিন্তু এই সিস্টেমে সেটা আছে অন্য একটা ফাইলে, যার নাম '/etc/shadow'। বলতে পারবেন, এই 'shadow' ফাইলটা আছে কোন ডিরেক্টরিতে? সেই ডিরেক্টরিটা আছে কোন ডিরেক্টরিতে? এই '/etc/shadow' ফাইলটা আবার সাধারণ ব্যবহারকারীর দেখার অনুমতি নেই। এখানে আপনাদের দেখাব বলে ফাইলটার কয়েকটা লাইন কপি করতে গেলাম। সিস্টেম করতে দিলনা। ব্যাটার সাহস ভাবুন, এদিকে যে ইলেকট্রিকে চলছে তার বিল দিই আমি। এখুনি সুইচ অফ

করে দিলেই পুরো দাঁত ছিরকুটে পড়বে। কপিটা আমায় ঘুরপথে করতে হল। প্রথমে সুপারইউজার হলাম, 'su' কমান্ড দিয়ে। এন্টার মারার সঙ্গে সঙ্গে সিস্টেম আমার কাছে রুট পাসওয়ার্ড জানতে চাইল। দিলাম সেটা। সুপারইউজার মোড চালু হল। তারপর এই ফাইলটা কপি করলাম অন্য একটা জায়গায় অন্য একটা ফাইলে। তারপর সেই ফাইলটার মালিকানা সঁপে দিলাম আমাকে, মানে, 'dd' নামের ইউজারকে। কারণ, সেটাও তো কপি করেছে রুট, রুটেরই মালিকানা, তাই আমি স্বাভাবিক অবস্থায় সেটা কিছু করতে পারব না। মালিকানা বদলানোর কমান্ডটার নাম 'chown'। ব্যস্ত হবেন না, এগুলোয় আমরা পরে আসব, এখানে একটু বললাম আগের থেকে একটা আন্দাজ দিয়ে রাখার জন্যে। আর নিজে নিজে কৌতূহল মেটানোর কোনো উপায় জানেন নাকি আপনি? এক্ষেত্রে এই পুরো পাঁয়তাড়াটা কষা গেল, ব্যক্তিগত পিসি হওয়ায় এর সুপারভাইজর আর ইউজার দুজনেই একই লোক, মানে আমি। তাই রুট পাসওয়ার্ড আমি জানি। আমি যখন আমার নিজের কাজের জন্যে সিস্টেম ব্যবহার করছি তখন আমি ইউজার, কারণ যা বললাম, রুট হয়ে কাজ করা সবসময়ই অত্যন্ত বিপজ্জনক। যে কোনো ডিরেক্টরিতে যে কোনো ফাইলে রুট-এর যে কোনো কাজের পারমিশন থাকে বলে সিস্টেম যখন তখন ঘেঁটে ফেলতে পারে রুট, একটু কমান্ডের ভুলে। ইনস্টলেশনের সময়ই দেখবেন, মানা করে দেয়, ওহে খোকা, নিজের জন্যেও একপিস ইউজার আইডেন্টিটি বানিয়ে নাও। ফাইলদের তথা ডিরেক্টরিদের এই অনুমতি/মালিকানা মানে পারমিশন/ওউনারশিপ — এগুলো আমাদের পরে খুবই ভালো করে বুঝতে হবে। এখন জাস্ট বুড়ি ছুঁয়ে গেলাম।

পাসওয়ার্ডগুলো কিন্তু এখানেও তাদের অবিকল চেহারায় নেই, আছে সঙ্কেতবদ্ধ বা এনক্রিপ্টেড চেহারায়। নইলে আপনি এখুনি আমার পাসওয়ার্ড জেনে ফেলতেন। এই এনক্রিপশন ব্যাপারটা বেশ ইন্টারেস্টিং। রোজার বরকে টেররিস্টরা তুলে নিয়ে গেছিল, সেই বরের পেশা ছিল এই এনক্রিপশন ভাঙা, ডিক্রিপশন। ক্লদ শ্যাননের এই এলাকায় গবেষণার কথা লিখেছিলাম তিন নম্বর দিনে, মনে আছে? এনক্রিপশনের ক্রিয়াপদ্ধতি বা বা অ্যালগরিদমটা এমন যে, কোনো একটা টেক্সটকে এনক্রিপ্ট করে সঙ্গেতবদ্ধ আকার দেওয়াটা খুবই সহজ, কিন্তু উল্টোটা, মানে সঙ্কেতবদ্ধ চেহারা থেকে তার প্রারম্ভিক আকারে ফিরিয়ে আনা প্রায় অসম্ভব। ভয়ঙ্কর শ্রম ও সময় ব্যয় করেও নিশ্চিত ফলাফল দেওয়াটা খুবই কঠিন। সিস্টেমও তাই এখানে একটা কুপথ মানে শর্টকাট ব্যবহার করে। প্রথমবার যখন ঢোকানো হচ্ছে সিস্টেমে, পাসওয়ার্ডটাকে এনক্রিপ্ট করে নেওয়া হয়। এবার সেই এনক্রিপ্টেড চেহারাটাকে লিখে রাখে '/etc/shadow' ফাইলে। এবার, যখনই আমাকে লগ-ইন-এর সময় পাসওয়ার্ড দিতে হচ্ছে, সিস্টেম সেই পাসওয়ার্ডটাকে আবার এনক্রিপ্ট করে দেখে নিচ্ছে যে, '/etc/shadow' ফাইলে রাখা চেহারাটার সঙ্গে সেটা মিলছে কিনা। এনক্রিপ্টেড চেহারাটাকে ভেঙে স্বাভাবিকটার সঙ্গে মেলানোর সে কোনো চেষ্টাই করছেনা। কারণ, যা বললাম, সেটা প্রায় অসম্ভব। এখানে আর একটা ব্যাপার আছে — মূল পাসওয়ার্ড আর তার এনক্রিপ্টেড চেহারার মধ্যে কোনো আম-দুধ-মেশামেশি বা ওয়ান-ইজ-টু-ওয়ান করেসপন্ডেন্স নেই। ক এবং খ দুজনেরই এনক্রিপ্টেড চেহারা একই গ হতে পারে, যদিও এর উল্টোটা ঠিক নয়, মানে একই ক থেকে দুটো আলাদা এনক্রিপশন গ আর ঘ পেতে পারিনা। তার মানে এনক্রিপ্টেডকে ডিক্রিপ্ট করে একাধিক চেহারা পাওয়া যেতে পারে, কিন্তু মূলটাকে যতবারই এনক্রিপ্ট করিনা কেন, ওই একই চেহারা পাব প্রত্যেকবার। তাই লগ-ইনের সময় দেওয়া পাসওয়ার্ডটাকে স্যাট করে একবার এনক্রিপ্ট করে দেখে নিলেই সিস্টেমের কাজ খালাস। 'login' প্রোগ্রাম প্রথমে আমাদের দেওয়া ইউজারনেম-টাকে মিলিয়ে নেয় '/etc/passwd' থেকে। তারপর, দেওয়া পাসওয়ার্ডটাকে এনক্রিপ্ট করে মিলিয়ে নেয় '/etc/shadow' থেকে। দুটোর একটাও যদি না-মেলে, তখন কনসোলে ফুটিয়ে তোলে ওই বাণী, 'login incorrect'। যদি দুটোই মিলে যায় তখন শেলকে জাগিয়ে তোলে এবং কনসোলে ফুটিয়ে তোলে কমান্ড প্রম্পট। এখানে দেওয়া '/etc/passwd' ফাইল থেকে দেখুন সব কজন ইউজারেরই জন্য লেখা শেলটা হবে '/bin/bash', মানে, ব্যাশ নিজেও তো একটা প্রোগ্রাম, সেই প্রোগ্রামটা আছে '/bin' ডিরেক্টরিতে, সেখান থেকেই চালাতে হবে তাকে। ঠিকানাটায়, ডিরেক্টরির নাম 'bin', তার আগে একটা '/' আছে। এর মানে কী?

আমার সিস্টেমের '/etc/shadow' ফাইল থেকে চারজন ইউজারের চারটে লাইন তুলে দিলাম।

```
dd:pV9quSXzPFD76:12296:0:99999:7:::
atithi:7NLMeKZlDh442:12298:0:99999:7:::
manu:XJNkQuu8YRCSU:12298:0:99999:7:::
piu:Lkut3i/V2Phzw:12353:0:365:7:-1::
```

একবার সফলভাবে লগ-ইন করে গেলাম মানে এবার যে যে ডিরেক্টরিতে আমার অধিকার আছে সেখানে আমি কাজ শুরু করতে পারব, আমার অধিকারাধীন ফাইলদের আমি বদলাতে পড়তে বা চালাতে পারব। এখানে এই মালিকানা এবং অনুমতির সঙ্গে গ্রুপ বা দলের একটা ধারণা আছে, সেটায় পরে আসছি। আর শুধু একটাই কথা বলার আছে লগ-ইনের কাজ নিয়ে। সেটা এই যে, শুধু শেলকে জাগিয়ে তোলা বা কমান্ড প্রম্পটকে ফুটিয়ে তোলাই না, আরো একটা কাজ করে লগ-ইন। ভারি পুণ্যের কাজ, পথহারা দিশাহারা মানুষকে তার বাড়িতে পৌঁছে দেওয়া, নিজের ঘরে বসিয়ে দেওয়া। প্রথম প্রথম একটা গ্নু-লিনাক্স সিস্টেমে ঢুকে যার দিশাহারা লাগেনা, এমন মানুষের স্বপ্ন দেখা খুবই স্বাস্থ্যকর অভ্যেশ — ওইসব স্বপ্ন থেকেই পৃথিবী বদলায়, বাস্তবতা বদলায়, বিপ্লব হয়, প্লেগের ওষুধ আবিষ্কার হয়। কিন্তু আমাদের আম-জনতাদের কাছে লগ-ইন একজন পরিত্রাতার মত — যাক বাবা, নিজের ঘরেই আছি। ইউজারকে তার নিজের নিজের হোমে, নিজের এলাকায় পৌঁছে দিয়ে যায় 'login'।

লগ-ইন হয়ে গেছে, কমান্ড প্রম্পট ফুটে উঠেছে, এখন যদি জানতে চাই, ফাইলসিস্টেমের ডিরেক্টরিজালের শাখা-প্রশাখার ঠিক কোথায় আছি, তার কমান্ড '`pwd`', প্রিন্ট-ওয়ার্কিং-ডিরেক্টরি। লিখে এন্টার মারা মাত্র, আমার বেলায়, কমান্ড প্রম্পটে ফুটে ওঠে, '/home/dd' — আমার ঘর। গ্নু-লিনাক্স সিস্টেমে '/home/dd' হল 'dd' নামের ইউজারের বাস্তুভিটে। 'dd'-র মালিকানার এলাকা। ইউজার 'manu'-র বেলায় সেটা হবে '/home/manu'। এভাবেই ঘটে প্রতিটা ইউজারের। সবারই নিজের ঘর ওই '/home' ডিরেক্টরিতে তার নিজের নামের একটা সাবডিরেক্টরি। অর্থাৎ 'login' ইউজারকে পৌঁছে দেয় তার নিজের ঠিকানায়। এবার সে '`ls`' কমান্ড লিখে এন্টার মারলে, মানে ফাইল ডিরেক্টরির লিস্ট দেখাতে বললে, নিজের হোমে নিজের শাসনাধীন এলাকায় নিজের মালিকানার ফাইল আর ডিরেক্টরিদের তালিকা দেখতে পায়। এবার '/etc/passwd' ফাইলের শেষ লাইনটা তুলে একটু বোঝার চেষ্টা করি।

```
piu:x:503:100:Smita Bhadra:/home/piu:/bin/bash
```

এর প্রথমটা, 'piu' হল ইউজারের নাম, যে নাম দেওয়া আছে সিস্টেমে, যে নামে সে লগ-ইন করে। প্রথা অনুযায়ী এটা ছোট হাতের অক্ষরেই হয়। এর পর যতিচিহ্ন, ':'। এখানে যে কটা ফিল্ড বা আলাদা আলাদা ক্ষেত্র আছে, তাদের প্রত্যেকটাকেই পরেরটা থেকে আলাদা করা হয়েছে এই যতিচিহ্ন ':' দিয়ে। শুধু এই ফাইল নয়, আরো বহু ফাইলেই এই যতিচিহ্ন ব্যবহার হয়, পরে দেখবেন। পরের ফিল্ডটা 'x', একটা প্রতীক যা পাসওয়ার্ডের উপস্থিতিকে চিহ্নিত করছে। পাসওয়ার্ডটা বা তার সাঙ্কেতিক চেহারাটা কিন্তু এখানে নেই, আছে '/etc/shadow' ফাইলে। তৃতীয় ফিল্ডটা, '503', ইউজারের ব্যক্তি-পরিচিতি (**U**ser-**Id**entity — uid)। এই ইউআইডি প্রত্যেকের আলাদা আলাদা। তার পরেরটা, '100' হল তার দলপরিচিতি (**G**roup-**I**dentiy — gid)। এই ফাইলে শেষ চারজনেরই 'gid' এক। তার পরের ফিল্ডটা, 'Smita Bhadra', ইউজারের পুরো নাম। এর পরের ফিল্ড, '/home/piu' হল এই ইউজারের হোম-ডিরেক্টরি। শেষ ফিল্ডটা, '/bin/bash' হল তার শেলের নাম, গ্নু-লিনাক্সের ডিফল্ট শেল ব্যাশ। মানে, আলাদা করে না-বদলালে সিস্টেম এটাকেই একজন ইউজারের শেল বলে ধার্য করে নেয়। ইউআইডি বা জিআইডি, এবং অন্য ডিটেইলগুলোয় আমরা পরে আসব ব্যবহারকারীদের অনুমতি এবং অধিকারের আলোচনায়। জাস্ট একটু কৌতূহল হিশেবেই এখানে আমি এই পাসওয়ার্ড ফাইলের ম্যানুয়াল পাতা থেকে একটা কুচো অংশ একটু তুলে দিচ্ছি। গ্নু-লিনাক্সে যে কোনো কমান্ডেরই ম্যানুয়াল পড়তে হয় ম্যান (**MAN**ual — man) কমান্ড দিয়ে, আগেই বলেছি। আর কোনো কনফিগারেশন ফাইলের ম্যানুয়াল পড়তে হয় '`man 5`' দিয়ে, কারণ, ম্যানুয়াল পাতাগুলোর ৫ নম্বর সেকশনে থাকে বিভিন্ন কনফিগারেশন ফাইলের হদিশ। এখানে আমি এই ফাইলটা পেতে গিয়ে যে কমান্ডটা দিয়েছিলাম সেটা হল, '`man 5 passwd > passwd.text`'। অর্থাৎ, ম্যানুয়ালের পাঁচ নম্বর সেট থেকে পাসওয়ার্ড ফাইলের পাতাগুলোকে একটা ফাইলে রূপান্তরিত করো যার নাম 'passwd.text'। আজকের একদম গোড়ার দিকের চার নম্বর সেকশনের আলোচনার রিডাইরেকশন বা চালান করার আগের উদাহরণটার সঙ্গে মেলান। শুধু '`man 5 passwd`' লিখে এন্টার মারলে যে পাতাগুলো ও আমায় স্ক্রিনে দেখাত, সেই পাতাগুলোকে রিডাইরেক্ট করে দিল, চালান করে দিল এই 'passwd.text' ফাইলে। আমি সেই ফাইল থেকে আমি একটা অংশ এখানে তুলে দিলাম, বিভিন্ন ফিল্ডগুলো কী ভাবে থাকে তারই নির্দেশটা, এর আগেই '/etc/passwd' ফাইল নিয়ে যে আলোচনাটা করলাম আমরা, তার সঙ্গে মিলিয়ে পড়ুন। এর থেকে একটু ম্যান পড়ার অভ্যেসও হবে। ম্যান পড়ার ব্যাপারটা গ্নু-লিনাক্সে এমন বিরাট জায়গা নিয়ে থাকে যে আগে চ্যাংড়ামো করে লিনাক্সকে আরটিএফএম (**R**ead-

The-Fucking-Manuals) সিস্টেম বলা হত। গ্নু-লিনাক্সের ব্যাপক সুবিধে এটাই। সবকিছু সমস্তকিছু জানার উপায়ই দেওয়া আছে সিস্টেমের মধ্যেই।

```
PASSWD(5)
NAME
       passwd - The password file
DESCRIPTION
       _passwd_  contains  various  pieces  of information
for each user account.
Included is
             Login name
             Optional encrypted password
             Numerical user ID
             Numerical group ID
             User name or comment field
             User home directory
             User command interpreter
The password field may not be filled if shadow passwords
have been  enabled.
If shadow passwords are being used, the encrypted
password will be found in _/etc/shadow_.
```

১০।। সিস্টেম মানে জাগ্রত এবং ঘুমন্ত প্রসেসের চিড়িয়াখানা

প্রথম যেবার টপ (top) কমান্ডটা ব্যবহার করলাম, আমার ভারি একটা অপরাধবোধ হয়েছিল। টপ কমান্ডটা নিরবচ্ছিন্ন ধারাবাহিক ভাবে একটা জ্যান্ত তালিকা দিতে থাকে স্ক্রিনে। জ্যান্ত তালিকা মানে বাস্তব সময়ের সঙ্গে সঙ্গে বদলাতে থাকে। কী কী প্রসেস বা প্রক্রিয়া সেই মুহূর্তে চলছে সিস্টেমে, তারা কতটা সিপিইউ ব্যবহার করছে, কতটা মেমরি ব্যবহার করছে, কতটা সময় ধরে চলছে, সেই তালিকাটা দেখায় টপ। এবং তালিকাটা মুহূর্মুহূ বদলাতে থাকে স্ক্রিনে। একটা প্রসেস একে ছিল, চারে নেমে যাচ্ছে, চারেরটা দুইয়ে উঠে যাচ্ছে, এইরকম চলতেই থাকে। টপ চালানোর আগে বহুক্ষণ ধরে সেদিন চুপচাপ কম্পিউটারের সামনে বসে বসে ঝিমোচ্ছিলাম, যেমনটা আমি প্রায়ই করে থাকি, যদিও স্বীকার করিনা সচরাচর। এক নম্বর ভারচুয়াল টার্মিনালে (tty1) কী একটা ম্যান পেজ খোলা ছিল আর ব্যাকগ্রাউন্ডে দু নম্বর টার্মিনালে (tty2) এমপ্লেয়ারে নুসরতের একটা এমপিথ্রি বাজছিল, আসলে কিছুই করছিলাম না, হঠাৎ শখ হল, টপটা চালালাম তিন নম্বর টার্মিনালে (tty3) গিয়ে। কী ভয়ঙ্কর — আমার এই কিছুই না-করে ঝিমোনোটাও আহা অপারেটিং সিস্টেম বেচারার এই বীভৎস কর্মকাণ্ডের বিনিময়ে। চলতে থাকা প্রসেসের মোট তালিকাটা বিকট, অগণ্য বললেই ঠিক হয় মোট সংখ্যাটা। আপনি করে দেখুন। যদি না-চমকান, কথা দিচ্ছি, কদিন ধরে হিং-গোলমরিচ-আদা দিয়ে একটা মুরগির রেসিপি মাথায় ঘুরছে, কাউকে আগে একটা খাইয়ে দেখতে হবে, মরে টরে গেল কিনা, তবে নিজে খাওয়ার রিস্ক নেব, আপনাকে ফ্রিতে খাইয়ে দেব। এই অগণ্য প্রসেসের অপরিমেয় জটিলতাকে নিজেদের মাথায় ধারণ করার থেকে আমাদের বাঁচিয়ে দেয় শেল। গ্নু-লিনাক্সের এটা একটা ভারি গুরুত্বপূর্ণ বৈশিষ্ট্য যে কারনেল আর শেল এখানে সম্পূর্ণ স্বতন্ত্র দুটো প্রোগ্রাম। শেল আর কারনেলের এই সম্পূর্ণ আলাদা দুটো প্রোগ্রামের মধ্যে একটা জীবন্ত সংযোগও থাকে — সিস্টেম কল হল সেই যোগাযোগ। মনে পড়ছে সিস্টেম কলের আলোচনা, দুই নম্বর দিন থেকে? গ্নু-লিনাক্সে একাধিক শেল থাকতে পারে — আলাদা আলাদা প্রয়োজন এবং পছন্দ মোতাবেক — ব্যাশ শেল, সি শেল, কর্ন শেল, জি শেল ইত্যাদি। পরে আমরা আসব এইসব নিয়ে বিশদ আলোচনায়।

বুটের পুরো কর্মকাণ্ডটা এবার মনে মনে ঝালিয়ে নিন — কী ভাবে প্রথমে বায়োস, তারপর লিলো, তারপর ইনিট, তারপর কারনেল, তারপর গেটি, তারপর লগ-ইন পরপর তাদের কাজ শুরু এবং শেষ করে, শেলের হাতে পতাকা হ্যান্ডওভার করে, একে একে সিন থেকে কেটে যায় প্রত্যেকেই। সিনে মানে স্ক্রিনে রয়ে যায় শুধু শেল, মানে কনসোলের গম্ভীর কালো সিরিয়াস পর্দায় ফুটে থাকে কমান্ড প্রম্পট। এবার একটু বোঝার চেষ্টা করা যাক, স্ক্রিনের পিছনে তখন জনান্তিকে কী ঘটছে। পাওয়ার সুইচ অন করার পর বায়োস থেকে শুরু করে এই কমান্ড প্রম্পটে আপনার কোনো কমান্ড টাইপ করে এন্টার মারা, মানে প্রোগ্রাম চালানো অব্দি গোটা ব্যাপারটাকে একটা জ্যান্ত

চিড়িয়াখানা বলে মনে হচ্ছে না? বোধহয়, একটুও ভুল মনে হচ্ছে না। মাল্টিপ্রোগ্রামিং অপারেটিং সিস্টেমের বিবর্তন নিয়ে ফ্রেড ব্রুকস-এর '৯৬-এ লেখা একটা বইয়ের প্রচ্ছদে ছিল একটা প্রাগৈতিহাসিক আলকাতরার খাদানে পড়ে যাওয়া ডাইনোসরদের ছবি — পর্দার পিছনে ঘটতে থাকা ইউনিক্স-এর জুরাসিক জু। বুট থেকে শুরু করে প্রোগ্রাম চালানো অব্দি এই চিড়িয়াখানার প্রতিটি প্রসেস প্রতীক্ষা করে রয়েছে — কখন তাদের হাতে কোনো কাজ দেওয়া হবে। অপেক্ষা করে আছে ইভেন্টের — ঘটনার — কমান্ডে বা শেলে বা কারনেলের সঙ্গে তাদের সিস্টেম কলে — কিছু একটা ঘটবে, যাতে তাদের প্রয়োজন পড়বে, ডাক পড়বে। এই ইভেন্ট বা ঘটনা মানে হতে পারে আপনি কোনো চাবি টিপলেন কিবোর্ডে, বা ইঁদুর নড়ালেন বা ইঁদুরের পেটে কোনো সুইচ টিপলেন, বা মেশিন যদি নেটওয়ার্কিত থাকে তাহলে হয়তো তড়িৎপথ বেয়ে কোনো তথ্য বা নির্দেশ বয়ে এলো দূরবর্তী কোনো মেশিন থেকে। ইভেন্ট ঘটা মাত্র কাজ শুরু করে দেবে সেই প্রসেস, ওই ধরনের ইভেন্টে যার কাজ শুরু করার কথা, কিন্তু যতক্ষণ-না ইভেন্টটা ঘটছে, প্রসেসরা কিন্তু মরে যায়নি, তারা সব ঘাপটি মেরে খাপ পেতে বসে আছে।

এই প্রসেসগুলোর সবচেয়ে বড় প্রসেস কিন্তু খোদ কারনেল নিজেই। সবচেয়ে বড় এবং সবচেয়ে বিশেষ। কারণ, সে আর যাবতীয় ইউজার প্রসেস বা ব্যবহারকারী পদ্ধতিগুলোকে নিয়ন্ত্রণ করে — যখন তারা সক্রিয় হয়ে ওঠে, শেষ অব্দি কাজ করার সুযোগ পায়, তাদের অপেক্ষমান অবস্থা ছেড়ে। এবং কারনেলই সেই একমাত্র পদ্ধতি যাকে সরাসরি হার্ডওয়ারদের সঙ্গে কথা বলতে হয়। ইউজার প্রসেসগুলো কারনেলের কাছে তাদের অনুরোধ জানায় তাদের প্রয়োজন মোতাবেক। ধরুন একটা প্রোগ্রামের কিছু কিবোর্ড থেকে আসা ইনপুট দরকার, সরাসরি ব্যবহারকারীর কাছ থেকে কিছু একটা এখুনি পেতে হবে তাকে, নইলে সে কাজ করতে পারছে-না। এবার সে ওই কিবোর্ড ইনপুটের যাচ্ঞা করে দরখাস্ত পাঠাবে কারনেলের কাছে। বা ধরুন তাকে স্ক্রিনে কিছু লিখতে হবে, মানে স্ক্রিনে কিছু আউটপুট পাঠাতে হবে। তার জন্যেও কারনেলের অনুমতি লাগবে। অনুমতি লাগবে যদি তাকে হার্ডডিস্ক থেকে কিছু পড়তে হয়, বা হার্ডডিস্কে কিছু লিখতে হয়। যদি তার মেমরি কাজে লাগাতে হয় তাহলেও দরখাস্ত দিতে হবে কারনেলকে। আর কাজ করতে গেলে এই প্রয়োজনগুলো তার প্রতিমুহূর্তেই গজিয়ে উঠবে। ভেবে দেখুন, কোনো ভৌত রসদ ব্যবহার না-করে একটা প্রোগ্রাম চলবে কী করে? চলা মানেই তো অন্তত মেমরিতে তোলা মূল প্রোগ্রামটাকে জাগিয়ে রাখা, তার মানেই মেমরির ব্যবহার। তাই প্রতিমুহূর্তে তাকে দরখাস্ত পাঠাতে হচ্ছে কারনেলের কাছে। মুহুর্মুহু এই অবিরত এবং নিরবচ্ছিন্ন দরখাস্তের স্রোতটারই অন্য নাম 'সিস্টেম কল'।

সাধারনত আইও বা ইনপুট-আউটপুট সংক্রান্ত প্রতিটা কাজই ঘটে কারনেলের মাধ্যমে। কারনেল সেই ব্যবস্থাটা করে যাতে একজন প্রসেসের আইও অন্য প্রসেসের আইও-র ল্যাজ মাড়িয়ে না দেয়। আপনার অ্যাকাউন্টিং সফটওয়ার যে বাইটগুলো পড়েছিল হার্ড ডিস্ক থেকে, ধরুন, আপনার সাবানের দাম, সেটা যাতে আপনার গ্রাফিক্স সফটওয়ারের খুলে রাখা ছবির ফাইলে দেগে না-দেয়। সামান্য দু-চারপিস ইউজার প্রসেসের কখনো সখনো একটু কারনেলের নিয়ন্ত্রণ কাটিয়ে পিছলে যাওয়ার অধিকার থাকে, কখনো সখনো তাদের সরাসরি আইও পোর্টে গিয়ে নিজের কাজ করার অনুমতি থাকে। এক্স-উইনডোজ লিনাক্সে কাজ করে একটা সার্ভার হিশেবে, এক্স উইনডোজ মানে আমরা তো আগেই বলেছি কমান্ডের জায়গায় ছবি দিয়ে তৈরি একটা প্রতিবেশ, যেখানে সব কিছুকে আমরা দেখি বুঝি বদলাই ছবি কাজে লাগিয়ে — এই এক্স উইনডোজ হল এই ধরনের পাঁকাল এবং পিছলে প্রসেসগুলোর মধ্যে প্রমুখ, যাকে প্রায়ই অধিকার দেওয়া হয় সরাসরি, কারনেল এড়িয়ে, আইও পোর্টে কিছু করার। যাইহোক, সে অনেক জটিল কথা।

অপারেটিং সিস্টেম আর ইউজার প্রোগ্রামগুলোর মধ্যে পারস্পরিকতা বা ইন্টারফেস হিশেবে এই সিস্টেম কলের ব্যবস্থা করে দেয় অপারেটিং সিস্টেমই। এক অপারেটিং সিস্টেম থেকে আর একটা অপারেটিং সিস্টেমে সবচেয়ে গুরুত্বপূর্ণ পার্থক্যের একটা হল এই সিস্টেম কলের গঠনের ফারাক। ধরুন, একটা বিশেষ অপারেটিং সিস্টেম ইউনিক্স — তার ফাইল সংক্রান্ত সিস্টেম কলগুলোকে ভাবছি আমরা — তাদের মধ্যে অনেকগুলো রকম আছে। ধরুন, একটা খুব সাধারন কাজকেই ভাবুন — একটা ফাইলকে একটা ডিস্কভূমি থেকে পড়ে আর একটা ডিস্কভূমিতে লেখা হচ্ছে। সচরাচর যাকে আমরা ফাইল কপি করা বলে ডাকি। এইটুকু একটা ছোট্ট কাজেও কিন্তু ব্যবহৃত হচ্ছে অনেকগুলো রকমের সিস্টেম কল। প্রথমে একটা সিস্টেম কল থাকবে ফাইলটাকে চেনানোর। ফাইলটার মধ্যে হার্ডডিস্কের শরীরের কত কত নম্বর সেক্টর থেকে কত কত কোন কোন বাইট তথ্য ভরা আছে, সেটা রয়েছে সেটা লেখা রয়েছে ফাইলসিস্টেমে। পরে আমরা দেখব ফাইলের ডিটেইলস লেখা এই তালিকাকে বলে আইনোড টেবিল।

এই আইনোড টেবিল থেকে ফাইলের খুঁটিনাটি মিলিয়ে হার্ডডিস্ক থেকে ফাইলটা পড়ার জন্যে একটা সিস্টেম কল থাকবে। একটা সিস্টেম কল থাকবে যা ঠিক করবে এবার হার্ডডিস্কের অন্য কোন জায়গায় এই বাইটগুলো নতুন করে লিখে ফেলা হবে, এবং লিখে ফেলবে বাইটগুলো। একটা সিস্টেম কল থাকবে কত গুলো বাইট অব্দি এই এক জায়গা থেকে পড়া আর অন্য জায়গায় লেখার প্রক্রিয়াটা চালিয়ে যেতে হবে। নতুন তৈরি হওয়া ফাইলটাকে ফাইলসিস্টেমের আইনোড তালিকায় আপডেট করে নেওয়ার জন্যেও একটা সিস্টেম কল থাকবে। ইত্যাদি।

আমরা আগে, চার নম্বর দিনে, ইউনিক্স-এর গঠনের ব্যাপারে যে আন্তর্জাতিক পোজিক্স স্ট্যান্ডার্ডের কথা বলেছি, সেই মোতাবেক সবগুলো ফ্লেভারের ইউনিক্স, মানে বেল ল্যাবের ইউনিক্স, সিস্টেম ভি, বিএসডি, গ্নু-লিনাক্স ইত্যাদি — এদের প্রত্যেকেরই সিস্টেম কলের গঠন মোটামুটি একরকম — সামান্য কিছু খুঁটিনাটির তফাত আছে। সিস্টেম কলের গোটা ব্যাপারটাই ভয়ানকভাবে মেশিন-নির্ভর — এখানে সফটওয়ারের ভূমিকা অত্যন্ত কম। এবং প্রায়শই এই সিস্টেম কলগুলো একটা অপারেটিং সিস্টেমে দেওয়া হয় অ্যাসেম্বলি কোডে লিখে। তারপর সি-থেকে তাদের ব্যবহার করার ব্যবস্থা করে দেওয়া হয়। সিস্টেমে এদের রাখা থাকে মূলত কিছু সিস্টেম কল ফাংশনের লাইব্রেরি হিশেবে। মানে সি থেকে কিছু নির্দিষ্ট ফাংশন ব্যবহার করা যায় যারা আসলে অ্যাসেম্বলি কোডে লেখা এই সিস্টেম কলগুলোকে কাজে লাগায়। শুধু সি নয় যদিও, অন্য অন্য কম্পিউটার ভাষা দিয়েও কাজে লাগানো যায় এই সিস্টেম কলগুলোকে।

এখানে একটা ব্যাপার মাথায় রাখুন, একটা এক-প্রসেসর কম্পিউটারের সিপিইউ বা সেন্ট্রাল প্রসেসিং ইউনিট একটা একক মুহূর্তে কেবলমাত্র একটি নির্দেশই পালন করতে পারে। কী করে সময়কে খুব ছোট ছোট টুকরোয় ভেঙে তার এক একটা টুকরোয় এক একটা নির্দেশ পালন করে মোটের উপর অনেকগুলো নির্দেশ একই সঙ্গে পালন করার একটা মায়া তৈরি করে কম্পিউটার তা আমরা মাল্টিপ্লেক্সিং-নিয়ে আলোচনায় আগেই বলেছি। তাই, এবার কোনো একটা নির্দিষ্ট ইউজার প্রোগ্রামের যদি কোনো সিস্টেম রসদ ব্যবহারের প্রয়োজন পড়ে, যেমন ডিস্কে লেখা কোনো ফাইল থেকে পড়া ইত্যাদি, সে তখন সিস্টেম কল নামক দরখাস্ত পাঠায় কারনেলের কাছে এবং তার নিজের কাজের নিয়ন্ত্রণ সেই মুহূর্তে ছেড়ে দেয় অপারেটিং সিস্টেমের হাতে। অপারেটিং সিস্টেম তখন তার প্রয়োজনের বিভিন্ন উপাদান আর মাত্রাগুলোকে বিচার করে সেইমত এই প্রোগ্রামের হয়ে ভৌত রসদটুকু ব্যবহার করে। তারপর ফের প্রোগ্রামের নিয়ন্ত্রণ ফেরত দেয় সেই ইউজার প্রোগ্রামের নিজের হাতে। সিস্টেম কলের কাজ শেষ হয়।

১১।। আর একটু খোসা চর্চা

বঙ্কিম বা রামকৃষ্ণ বা রাজশেখরের বালখিল্য পর্বতমুনি যতই খোসা-অন্তপ্রাণ হতে বারণ করুন, আপাতত আমাদের একটু খোসা-চর্চা করতেই হবে। আগেই বলেছি, এই খোসা বা শেলই সেই মারাত্মক মিডলম্যান, আমাদের কমান্ড ইন্টারপ্রিটার — আমাদের কমান্ড যে ইন্টারপ্রিট করে পৌঁছে দেয় কারনেলের কাছে — এবং কারনেল তখন আমাদের চালানো প্রোগ্রামগুলোর চাহিদা এবং প্রয়োজন মোতাবেক রসদ সরবরাহ করে। তার মানে, আর একভাবে দেখলে, শেল আমাদের কাছ থেকে কারনেলকে লুকিয়েও রাখে, অপারেটিং সিস্টেমের খুঁটিনাটি জটিলতাগুলোকে গোপনে জনান্তিকে অগোচরে রেখে দেয়। আমাদের আর মাথা ঘামাতে হয়না। এক নম্বর দিনে আমরা যে প্রতীকী বা সিম্বলিক ভারচুয়াল ভূমির কথা বলেছিলাম, ভৌত উপাদানগুলোর ভিত্তির উপর দাঁড়িয়ে কী করে স্তরে স্তরে সেটাকে বুনে তোলা হয় — এখন সেটা একটু একটু করে বুঝতে পারছেন?

আমাদের খোসা বা শেলও একটা ইউজার প্রসেস। সে আমাদের কিবোর্ডে চাবি টেপার, ইঁদুরের পেট কচলে দেওয়ার নানা ইভেন্টের বা ঘটনার জন্যে অপেক্ষা করে। এই অপেক্ষা করা মানে একমনে আইও পোর্টগুলো পড়তে থাকা — কিছু সেখানে ঘটল কিনা। এই ক্রিয়াটা গোটাটাই ঘটে কারনেলের মধ্যে দিয়ে। যেই কারনেল দেখে আপনি কোনো একটা কমান্ড লিখে এন্টার মারলেন — সেই এন্টার মারার আগের গোটা লাইনের টেক্সটা সে পাঠিয়ে দিল শেলের কাছে। শেল তার জানা কমান্ডগুলোর সঙ্গে এবার মিলিয়ে মিলিয়ে দেখে নিল আপনি যা যা টাইপ করেছেন সেই চাবিগুলোকে। ধরুন আমরা টাইপ করেছি '`ls`', তারপর এন্টার মেরেছি। কারনেল এন্টারের আগেরটুকু মানে 'ls' অংশটাকে পাঠিয়ে দিয়েছে শেলের কাছে। শেল এবার মিলিয়ে দেখতে শুরু করে দিয়েছে, এই 'ls' আকারের কোনো কমান্ড আছে কিনা যাকে সে চেনে। পরে আমরা দেখব, শেলের এই খোঁজাটা যে গোটা মেশিন মানে হার্ড ডিস্ক জুড়ে

ঘটে তা কিন্তু নয়। নির্দিষ্ট কিছু ডিরেক্টরিতে খুঁজে দেখে শেল। যে ডিরেক্টরিগুলোর নাম তার পাথ বা পথনির্দেশ-এ দেওয়া আছে। শেলের কাছে দেওয়া এই পথনির্দেশে কী কী ডিরেক্টরি রয়েছে সেটা আমরা জানতে পারি 'echo' কমান্ড ব্যবহার করে। আর, পথনির্দেশ বলে সিস্টেমে যেটা থাকে, তার টেকনিকাল পরিচিতি হল একটা '$PATH' নামে একটা শেল ভ্যারিয়েবল। অনেক কিছুই এখানে বোঝার আছে। নয় আর দশ নম্বর দিনে আসব এই প্রসঙ্গে। আমরা কমান্ড দেব, 'echo $PATH'। এরপর এন্টার মারলেই শেল আমাদের দেখাবে, কী কী ডিরেক্টরিতে তার খুঁজে দেখার কথা। যারা উইন্ডোজ প্রতিবেশে 'autoexec.bat' ফাইলটাকে কখনো কাজে লাগিয়েছেন তারা দেখবেন এর মধ্যে দেওয়া থাকে উইন্ডোজ ওএস-এর পথনির্দেশটা। উইন্ডোজ প্রতিবেশে পথনির্দেশ জানতে হলে ডস-প্রম্পটে কমান্ড দিতে হয়, 'path'।

গ্নু-লিনাক্সে পথনির্দেশের কাঠামোটার সঙ্গে একটু পরিচিত হয়ে নিন এবার। আমার গ্নু-লিনাক্স সিস্টেমে আমি সাধারণ ইউজার হয়ে, মানে, 'dd' হয়ে, 'echo $PATH' কমান্ড দেওয়ায় স্ক্রিনে ফুটে উঠল এই রকম কিছু —

```
/home/dd/bin:/usr/local/bin:/usr/bin:/usr/X11R6/bin:/bin:/usr/games:
```

আবার 'su' করে রুট পাসওয়ার্ড দিয়ে রুট হয়ে ওই একই কমান্ড দেওয়ায় ফুটে উঠল —

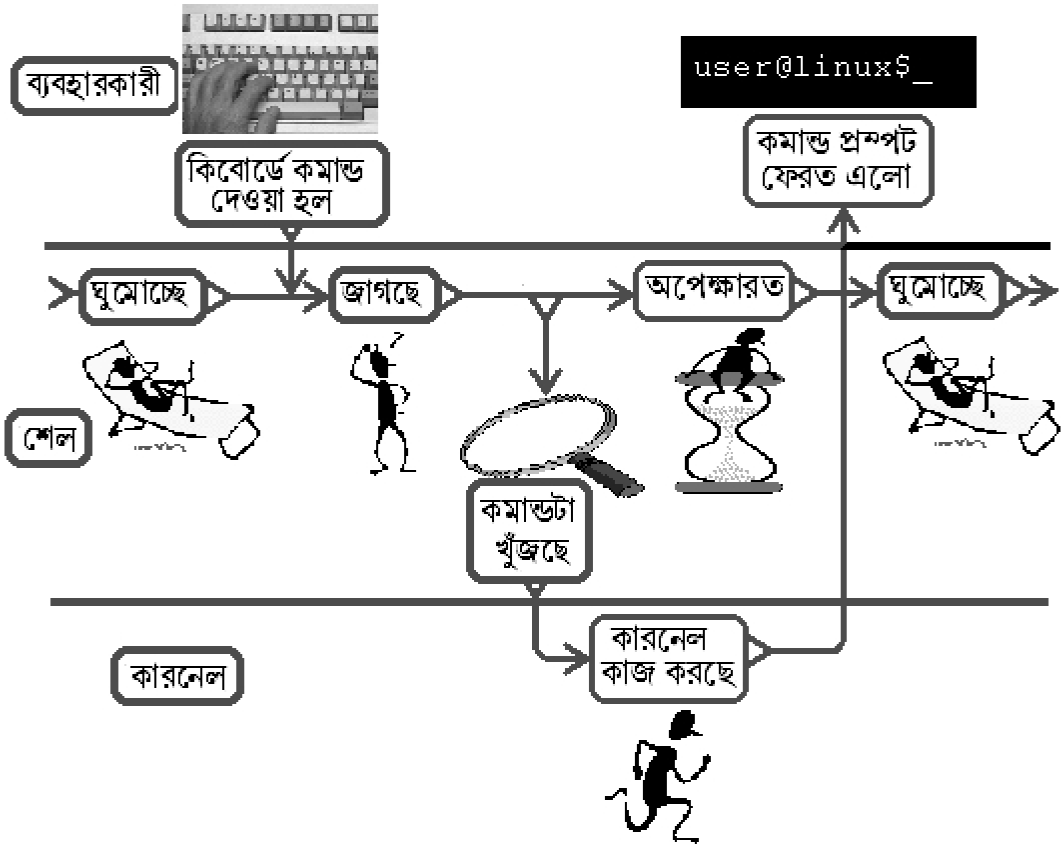
```
/sbin:/usr/sbin:/usr/local/sbin:/root/bin:/usr/local/bin:/usr/bin:/usr/X11
R6/bin:/bin:/usr/games
```

বাইনারি ফাইল মানে, আগেই বলেছি, যে ফাইলগুলোর নাম লিখে এন্টার মারলে নিজে নিজেই চলে। যেমন, 'ls' একটা বাইনারি, কারণ, 'ls' লিখে এন্টার মারলে সে নিজে নিজেই কাজ করে এবং ফাইল আর ডিরেক্টরির তালিকা ফুটিয়ে তোলে। উইনডোজ প্রতিবেশে '*.exe' ইত্যাদি এক্সিকিউটেবল ফাইলগুলো যা করে। 'dd'-র হোম ডিরেক্টরি '/home/dd'। সেখানের বাইনারি ডিরেক্টরি হল '/home/dd/bin', 'dd'-র ব্যবহারযোগ্য বাইনারি ফাইলগুলো যেখানে থাকে, সেটা রয়েছে 'dd'-র পথনির্দেশে। আর রুটের হোম '/root', সেখানের বিন ডিরেক্টরি '/root/bin', সেটা আছে রুটের পথনির্দেশে। এর বাইরে রুটের পথনির্দেশে আছে তিনটে আলাদা সিস্টেম বাইনারির মানে সিস্টেম প্রোগ্রামের ডিরেক্টরি, '/sbin:/usr/sbin:/usr/local/sbin'। এগুলো 'dd'-র নেই, কারণ ওই কাজগুলো করার, ওই প্রোগ্রামগুলো চালানোর অনুমতি সাধারণ ইউজারের থাকেনা। আর সেই ':' চিহ্নটা এখানেও ব্যবহার হচ্ছে যতিচিহ্ন হিশেবে।

শেলের কথায় ফেরত আসা যাক। আমি 'ls' লিখে এন্টার মেরেছি। শেল খোঁজা শুরু করেছে সেই ডিরেক্টরিগুলোয় এই মাত্র 'dd'-র 'echo $PATH' কমান্ড থেকে যে পথনির্দেশ পেলাম। এই পথগুলোর মধ্যে যদি এবার শেল 'ls' নামে কোনো বাইনারি ফাইল পেয়ে যায়, সেই 'ls' ফাইলটাকে চালিয়ে দেয়। এবং তার আউটপুট মানে ফলাফলটাকে পৌঁছে দেয় স্ক্রিনে। যাতে 'ls' কমান্ড চালানো ইউজার সেটা দেখতে পায়। এই ক্ষেত্রে, এই 'ls' নামের বাইনারিটাকে শেল পেয়ে যাবে '/bin' ডিরেক্টরিতে, যা পথেই আছে। পাওয়া মাত্র শেল কারনেলকে আবেদন জানায় একটা '/bin/ls' প্রসেস চালু করার। মানে, মোদ্দা কথায়, 'ls' প্রোগ্রামটা চালানোর। এই আবেদনটা জানাতে হল কেন? নিজেই ভাবুন তো? প্রতিমুহূর্তে নিজেকে জেরা করতে করতে এগোন। এখানে আবেদনটা জানাতেই হবে। 'ls' প্রোগ্রামটা চলা মানেই তো ডিস্ক থেকে ডিরেক্টরির আর ফাইলের তালিকা বানানো, তার মানেই ডিস্ক পড়তে হবে, তার মানেই ডিস্ক নামক ভৌত রসদের সঙ্গে কথা বলার, আর সেই কথা বলার অধিকার একমাত্র কারনেলের আছে। এবার এই '/bin/ls' প্রসেসটা একটা লিস্ট বা তালিকা তৈরি করে যে ডিরেক্টরি থেকে 'ls' কমান্ডটা দেওয়া হয়েছিল, সেই ডিরেক্টরিতে মোট উপস্থিত ডিরেক্টরি এবং ফাইলের। এই '/bin/ls' চালিয়ে তালিকা বানিয়ে সেই তালিকাটা স্ক্রিনে পাঠানোর দায়িত্বটা কারনেলকে দিয়ে শেল ঘুমোতে যায়। কেন, আবার কারনেল কেন? যদি না-বুঝতে পারেন, একটু গান শুনুন, পরে পড়বেন। যতটুকু সময় ধরে '/bin/ls' কাজ করে ততক্ষণ ঘুমোয় শেল। এই তালিকাটা এবার কারনেল পৌঁছে দেয় ইউজারের কাছে। কনসোল একটা ভৌত রসদ, তাই, এটা আর শেলের এলাকা নয়, কারনেলের কাছে সিস্টেম কল নামক দরখাস্ত করাটুকুই তার কাজ।

'/bin/ls' প্রক্রিয়াটা শেষ হয়, শেষ হওয়ার সংবাদ কারনেলের কাছে পৌঁছে দেয় '/bin/ls' নিজেই। এটা প্রোগ্রামেরই অংশ — একজিট (exit) নামক সিগনাল কারনেলের কাছে পৌঁছে দেওয়া। এই সিগনাল হাতে পেয়ে কারনেল এবার শেলকে ফের জাগিয়ে তোলে। আবার জেগে থাকতে বলে। জেগে থাকা মানে অতন্দ্র প্রহরা — কিবোর্ডে কিছু

নতুন কমান্ড এল কিনা। বা মাউসে। কোনো নতুন ঘটনা। শেল কারনেলের আদেশ মোতাবেক জেগে ওঠে, আবার স্ক্রিনে ফুটে ওঠে কমান্ড প্রম্পট। যার মানে শেল ফের অপেক্ষা করছে নতুন কমান্ড টাইপ করে এন্টার মারার। আমরা স্ক্রিনে দেখলাম ‘ls’ টাইপ করার পর এন্টার মারা মাত্রই কমান্ড প্রম্পট উঠে গিয়ে স্ক্রিনে ফুটে উঠল ফাইল আর ডিরেক্টরির তালিকাটা। তারপর ফের ফেরত এল কমান্ড প্রম্পট। মানে ‘ls’ আদেশটা পালন করা হয়ে গেছে।



এবার যদি কোনো একটা স্টেপে কিছু ঘেঁটে যায়? বা অকারণ দেরি হয়? ধরুন, যে ডিরেক্টরিতে ‘ls’ মেরেছেন সেটা বেধড়ক লম্বা, কয়েক কোটি ফাইল আর ডিরেক্টরি আছে সেখানে, এবং আপনার প্রসেসরও একটু মাঠো, মার্কিন কায়দায় বললে ‘স্টেট-অফ-দি-আর্ট’ নয় মোটেই, সেক্ষেত্রে কমান্ড প্রম্পট তো ফিরে আসছে না, দেরি করছে — আপনি এখন কী করবেন? আগেই বলেছি, কন্ট্রোল আর অল্ট সুইচদুটোকে টিপে রেখে এফ-ওয়ান থেকে এফ-সিক্স অব্দি যেকোনো একটা এফ-সুইচ টিপলেই আপনি সেই নম্বর ভারচুয়াল কনসোলে চলে যাবেন। ম্যানড্রেক বা সুজে ডিস্ট্রিবিউশনে এই কাজটা উইন্ডোজ চাবিটা টিপেই করা যায়, মানে গলে যেতে থাকা বা উড়ে যেতে থাকা উইনডোজের লোগো আঁকা সুইচ-দুটো। ডানদিকেরটা টিপলে পরের ভারচুয়াল কনসোলে আর বাঁদিকেরটা টিপলে আগের ভারচুয়াল কনসোলে যাওয়া যায়। অন্য কনসোলে গিয়ে অন্য কোনো একটা কাজ দিন মেশিনকে।

উইন্ডোজ থেকে যারা এসেছেন তাদের কাছে এটা সত্যিই একটা নতুন অভিজ্ঞতা — সত্যিকারের মাল্টিটাস্কিং কাকে বলে দেখুন। উইন্ডোজ নিজেকে মাল্টিটাস্কিং ওএস বলেই দাবি করে। কিন্তু সেই মাল্টিটাস্কিং-টা উপর থেকে চাপানো। সেটা বোঝা যায় গ্নু-লিনাক্স করতে গেলে। এমনও হয়েছে আমার সিস্টেমে যে, একটা কনসোলে প্রম্পট আর সেদিনের মত ফেরতই আসেনি। অথচ আমি অন্য কনসোল থেকে এক্স-উইন্ডোজ অব্দি চালিয়েছি। সেখানে অন্যান্য কাজ করে গেছি। পরের বার যখন বুট করেছি তখন সেই হারিয়ে যাওয়া কনসোলটাকে ফেরত পেয়েছি। আমরা যারা উইন্ডোজে কাজ করেছি অনেকগুলো বছর, তাদের বেলায় এই অভিজ্ঞতাটা বেশ ব্যাপক। গ্নু-লিনাক্স তথা যে কোনো ইউনিক্সের শেল নামক বস্তুটা ভীষণ রিজনেবল। আমি এই রিজন ঘেঁটে যেতে দেখিনি কখনো। উইনডোজে বেশিদিন একটানা কাজ করার পরই নিজের বুদ্ধি এবং যুক্তিবোধ দুটোর উপরই ভরসা কমে যেতে থাকে, কেন কোনটা কখন কী হয় — এটা প্রায় জীবনরহস্যের মতই অবোধ্য লাগে।

এখানে আমার নিজের একটা মজা পাওয়ার গল্প করে নিই। আমার মেশিনে একটা ব্যাক-আপের কাজ চলছিল। আমাদের কলকাতা লাগ-এর ইন্দ্র আর তথাগত এসেছিল আমার একটা ঘেঁটে যাওয়া স্কাসি কনফিগারেশন ঠিক করার চেষ্টার নাম করে আড্ডা মারতে আর কাবাব ইত্যাদি খেতে। হঠাৎ কারেন্ট চলে গেল। ইউপিএস-টা চলার কথা পনেরো মিনিট, বারো সাড়ে বারোর বেশি চলেনা। অথচ ব্যাক আপে তার চেয়ে বেশি সময় লাগবে। কিছুটা বিদ্যুৎ বাঁচানোর প্ল্যানে তথাগত কিবোর্ডে টাইপ করল ‘পাওয়ারঅফ’ (poweroff) এবং এন্টার মারল, স্ক্রিনে তখন লাইনের পর লাইন ব্যাকআপশীল ফাইলের তালিকা হুড়হুড় করে নেমে যাচ্ছে, তারপর মনিটরটা বাইরে থেকে অফ করে দিল। মানে পাওয়ারঅফ কমান্ডটা সত্যিই মেশিনের কাছে পৌঁছল কিনা আমরা তা দেখতে পেলাম না। এই পাওয়ারঅফ কমান্ডটা তার নাম অনুযায়ী সত্যিই পাওয়ারকে অফ করে। একসময় মেশিন থেমে গেল। এবং পরে কারেন্ট এলে বুট করে দেখলাম সত্যিই ব্যাকআপের কাজ শেষ করে মেশিন অফ হয়েছিল। মানে আগের কমান্ডটা শেষ হওয়ার পরে প্রম্পট ফেরত এসেছিল, মানে আসেনি কারণ কনসোল তখন অফ ছিল, এবং সেখানে চালু হয়েছিল পাওয়ারঅফ কমান্ডটা এবং তার পরের এন্টার। যে অংশটা ততক্ষণ মেশিনের বাফারে ছিল। এটা এমন কিছু ঘটনা না, কিন্তু আমি ভারি আমোদ পেয়েছিলাম। গ্নু-লিনাক্সে এটা করা চলে। যতক্ষণ খুশি যতবড় খুশি কমান্ড টাইপ করে চলা যায়। আমি স্ক্রিনে হয়ত দেখতে পাচ্ছিনা, কিন্তু তাতে কাজের কোনো অসুবিধা হয়না। আমরা আগেই উল্লেখ করেছি এটাই ইউনিক্স-এর ফুল-ডুপ্লেক্স সিস্টেমের সুবিধে। কিন্তু এখন না এসব, পরে আসবে, আমরা যখন আরো বড় করে ইনপুট আউটপুটের আলোচনায় আসব।

বুট প্রক্রিয়ার প্রাথমিক পরিচয়টা আমরা পেলাম, এবং শেলের ক্রিয়াপদ্ধতির। পরের দিন আমরা হার্ডডিস্কের পার্টিশনিং এবং ফাইলসিস্টেমকে নিয়ে এই বুটপ্রক্রিয়াটাকে আরো একটু বিশদে বুঝব। এবং কিছু কমান্ডকেও। শেষ অব্দি সরাসরি লিনাক্সে আমরা পৌঁছলাম, এখনো বিশ্বাস হচ্ছে না, যেরকম ঘুরপথের পর ঘুরপথ চলছিল। পাঠমালাটা পড়া এবং তার কমান্ডগুলোকে করে করে দেখা কিন্তু চলতেই থাকবে। নইলে মাথায় বসবে না। পড়ার জায়গাগুলো মোটামুটি একই আছে। সুমিতাভ দাস, কারনিঘ্যান পাইক, স্ট্যানফিল্ড আর স্মিথের ‘লিনাক্স — সিস্টেম অ্যাডমিনস্ট্রেশন’, তার সঙ্গে ইন্ট্রোডাকশন টু লিনাক্স বলে মাখটেল্ট গারেলস-এর একটা বই, জিএলটির সিডিতে আছে, একটু ওল্টানো যেতে পারে, মানে, ইয়ে, পেজ আপ ডাউন করা যেতে পারে।

glt-mad@ilug-cal.org

MADHYAMGRAM

গ্নু-লিনাক্স ইশকুল

জিএলটি মধ্যমগ্রাম
glt-mad@ilug-cal.org

আগেও বলেছি, আরো একবার মনে করিয়ে দিই, গ্নু-লিনাক্স ইশকুলের এই পাঠমালাটা কোনো টেক্সট বই না, এটা পড়ে আপনি কিছুই শিখবেন না, শিখবেন কী করে সেটা শেখার চেষ্টা করা যেতে পারে, বড়জোর। কম্পিউটার, তাও আবার গ্নু-লিনাক্সের মত কোনো জ্যান্ত গতিশীল এবং সর্বব্যাপী বিষয় নিয়ে, আমার যোগ্যতার কয়েক কদম বাইরে। আর টেক্সট বই ব্যাপারটার প্রতি একটা অ্যালার্জি আছে আমার জন্মের থেকে, এবং নিয়েই মরতে যাব বলে বিশ্বাস। আমি নিজে গ্নু-লিনাক্সে উত্তেজিত হয়েছিলাম, যখন এলোপাথাড়ি শেখার চেষ্টা করে চলেছিলাম, তখন চিন্তার কিছু কাঠামো কিছু খোঁচ কিছু ভাঙ্চুর কিছু নির্মাণ নিজের কাছে খুব গুরুত্বপূর্ণ লেগেছিল, যে এলাকাগুলো আর কাউকে, যেও একইভাবে, একই রকম গোলগোল এবং অশিক্ষিত উপায়ে চেষ্টা চালাচ্ছে, সাহায্য করতে পারে নিজের চিন্তাকে নিজের কাছে গুছিয়ে নিতে, নিজের চেষ্টার সঙ্গে নিজেরই কথোপকথনে। এখানে যে লিখছে যে পড়ছে তাদের মধ্যে কমন হল ওই অপারগতা এবং গোলগোল অশিক্ষা। আর অপরে মার্জিন অফ মার্জিন নিয়ে লেখার পর থেকেই এই ইনফর্মাল লেখার প্রতি আমার একটা ভালো লাগা তৈরি হয়, আমার অনেক বন্ধুরই সেটা কাজে লেগেছিল। পলিটিকাল ইকনমি আর সংস্কৃতি স্টাডিজ হয়ত আমার নিজের জায়গা, কম্পিউটারটা নয়। কিন্তু সেই না-হওয়াটাই এই বইটার শর্ত, যেমন সত্যিকারের গরিব ঘরের কষ্টে থাকা মেয়েদের, এলিট ঘরের ফেমিনিজম বেচা সেলসগার্লদের কথা বলছি না, কিছু জায়গা থাকেই যা অন্য একটা মেয়ের জায়গা থেকে বোঝা বেশ সহজ, তেমনি আপনার আর আমার একদেশতার সূত্র এটাই — দুজনেরই না-জানা, এবং জানতে-চেষ্টা-করা। এর যেকোনো একটা শর্তকে তুলে নিলেই এই লেখাটা জাস্ট হেজে যায়। আমার খোঁজাটাকে আমি আপনার সঙ্গে শেয়ার করছি, আমি চাইছি চিন্তার ধাঁচটাকে আমি যেভাবে পেয়েছি আপনার কাছে পৌঁছে দিতে। এটা আপনার উপরেও একটা শর্ত চাপিয়ে দিচ্ছে, যদি নিজের কোনো খোঁজা এবং চেষ্টা, এই লেখা থেকে আলাদা করে, না-থাকে, আপনার এটা পড়ার কোনো মানে নেই। আপনি এটা পড়ে যে কিছুতেই গ্নু-লিনাক্স শিখবেন না, এ গ্যারান্টি আমি আগে থেকেই দিয়ে দিচ্ছি। আজ আমরা শুরু করব কয়েকটা অত্যাবশ্যক কমান্ড দিয়ে, তারপর যাব ফাইল সিস্টেমের আলোচনায়।

।। দিন ছয় ।।

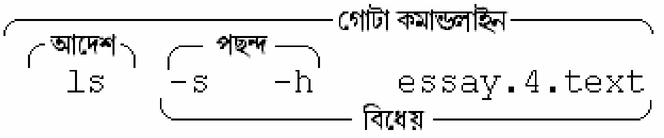
১ ।। কয়েকপিস কমান্ড

এবার একদম কেজো একটা জায়গা। পাঁচ নম্বর দিনে লগ-ইন করে আপনি ঢুকে পড়েছেন সিস্টেমে, সেখানে শেল এবং কারনেলের উপস্থিতি সম্পর্কেও একটা আলগা আন্দাজ তৈরি হতে শুরু করেছে। কিন্তু এবার সিস্টেমটাকে নিজের সামনে চলমান দেখতে গেলে কয়েকটা কমান্ড জেনে নেওয়া খুব জরুরি। আমরা কয়েকটা কমান্ডের উল্লেখ করেছি এর মধ্যেই এক নম্বর দিনে দু-চারটে, এবং পাঁচ নম্বর দিনে গোটাকয়েক। সেগুলোকেই এবার একটা প্রথানিষ্ঠ রকমে ঝালিয়ে নেওয়া যাক, হ্যাঁ দেখবেন, ধূপটা যেন না-নেভে, উঁহু আওয়াজ নয়, এখানে কমান্ডপাঠ হচ্ছে ভাই।

আমাদের অভ্যস্ত গ্নু-লিনাক্স কমান্ডের কাঠামোটা অনেকটা এইরকম — মোট কমান্ড লাইনটার দুটো অংশ, একটা মূল কমান্ড, আদেশটা। অন্যটা হল তার আর্গুমেন্ট, বিধেয়। এই বিধেয়-তে আবার দুটো অংশ, একটা অপশান বা পছন্দ। অন্যটা হল যার উপর আদেশটা চলবে সেই অবজেক্ট বা কর্মটা। ধরুন আপনি পাঁচ নম্বর দিনের আপনার ওই প্রবন্ধের বইয়ের ডিরেক্টরিতে ঢুকেছেন। একটু আত্মশ্লাঘা পেতে চাইছেন নিজের প্রবন্ধরাজির গাবদা গাবদা গতর দেখে। এই মুহূর্তে আপনি যে প্রবন্ধটা লিখছেন তার নাম essay.4.text — আগেরগুলো শেষ হয়ে গেছে। অবভিয়াসলি, আমারও তাই। পরশু প্লাস্টার কাটবে, ঠ্যাং ছুটি শেষ হয়ে এল, আজ নয়ই ডিসেম্বর, এখন প্রায় যুদ্ধকালীন তৎপরতায় লিখতে হচ্ছে।

এবার, আপনি এর সাইজ জানবেন, কমান্ড দিলেন —

```
ls -s -h essay.4.txt
```



এবং এর পর এন্টার মারলেন, অহো, কী আনন্দ, বাইট সাইজ দেখছি ধ্যাড়াধ্ধ্যাড় করে বেড়ে যাচ্ছে, অহো, কী আনন্দ, মনে হচ্ছে এই ধরিত্রীতে একটা পদচিহ্ন না রেখে কিছুতেই মরব না, বিগলিত হয়ে গেলেন ফাইলের সাইজ দেখে। এবার কমান্ডের কাঠামোটা দেখুন। এর এই '`ls`' অংশটা হল মূল কমান্ড বা আদেশ, '`-s -h`' অংশটা হল অপশান বা পছন্দ, '`-s -h essay.4.txt`' এই পুরোটা হল আর্গুমেন্ট বা বিধেয়, এবং '`ls -s -h essay.4.txt`' পুরোটা মিলিয়ে হল গোটা কমান্ডলাইনটা। এখানে আপনার হয়ে পছন্দটা আমি করে দিয়েছি, বলে দিয়েছি কী অপশান হবে। '`s`' মানে সাইজটা দেখাও, আর '`h`' অপশানটা বলে দিচ্ছে ওই সাইজের একক করো বাইট বা কিলোবাইট বা মেগাবাইট এমন কিছু যা মানুষে বোঝে, হিউম্যান রিডেবল হয়, নয়তো ও সাইজটা দেবে ব্লক-এর সংখ্যা দিয়ে। ব্লক ব্যাপারটায় আসছি আমরা একটু বাদে। দুটো অপশানেরই আগে, দেখুন, আমরা

একটা করে '-' চিহ্ন দিয়েছি। এটা অনেক কমান্ডের সঙ্গে দিতে হয়, অনেক কমান্ডের সঙ্গে দিতে হয়না। অনেক কমান্ডের বেলায় আবার দুটো অপশানকে একসঙ্গে জুড়ে দেওয়া যায়। যেমন এই কমান্ডটা হুবহু একই ভাবে কাজ করত যদি আমরা কমান্ডলাইন দিতাম — `ls -sh essay.4.txt`। এবার প্রশ্ন হল, কোনটার উপর কমান্ডটা কাজ করবে, মানে কর্ম বা অবজেক্ট, এই উদাহরণে যা '`essay.4.txt`', সেটা নয় আপনি জানেন, এবং কী কমান্ড সেটাও জেনে ফেললেন, কিন্তু কোন কমান্ডে কী অপশান, কোন অপশান কী আকারে দেওয়া যায়, এই হাজারো খুঁটিনাটি আপনি কেমন করে জানবেন? জানবেন এই টেক্সট থেকে।

```
Usage: ls [OPTION]... [FILE]...
List information about the FILEs (the current directory by default).
-a, --all                do not hide entries starting with .
-h, --human-readable     print sizes in human readable format (e.g., 1K 3M 2G)
-i, --inode              print index number of each file
-l                       use a long listing format
-R, --recursive          list subdirectories recursively
-s, --size               print size of each file, in blocks
-t                       sort by modification time
-1                       list one file per line
    --help               display this help and exit
```

এটা অনেকটা ত্র্যম্বকের ত্রিনয়ন ত্রিকাল ত্রিগুণের মত হল, এই টেক্সটা গজাল কোত্থেকে? ধীরে। এই টেক্সটের গজানোর ইতিহাসটা এই টেক্সটের মধ্যেই আছে। দেখুন তো আলোচনাটা হওয়ার আগে সেটা নিজেই খুঁজে পান কিনা? পেলেন? প্রথম লাইনটা দেখুন, ব্যবহারবিধি বা ইউসেজ, ঠিক অপশান আর্গুমেন্ট কমান্ড-লাইন নিয়ে আমরা এর আগেই যা বললাম সেই কথাটাই অন্য আর এক রকমে বলা। দ্বিতীয় লাইনে বলা কী কাজ করে এই কমান্ডটা। তিন থেকে এগারো নম্বর অব্দি প্রতিটি লাইনের কাঠামোই এক। প্রথমে একটা অপশানের ছোট আকার, তারপর বড় আকার, তারপর সেই অপশানটা কী কাজ করে। ছোট আর বড় দুটোই দেওয়া যায়, ছোটটা দেওয়ার সুবিধে, আর বড়টা মনে রাখার। শুধু দেখুন ছয়, নয় আর দশ নম্বর লাইনের পছন্দের কোনো বড় আকার নেই, শুধু ছোটটাই আছে। আর এগারো নম্বর লাইনের পছন্দটার শুধু বড় আকারটাই আছে। এর মধ্যে তিন নম্বর লাইনের প্রথম অপশানটা বুঝতে গ্নু-লিনাক্সে কম পরিচিতদের একটু অসুবিধে হতে পারে। একটা ডিরেক্টরির সব ফাইলকে স্বাভাবিক অবস্থায় `ls` দেখায় না। যেসব ফাইল বা ডিরেক্টরির নামের আগে একটা করে '.', বিন্দু বা ডট আছে, যেমন ধরুন .kde বা .mplayer ইত্যাদি। সচরাচর এই ফাইলগুলোয় থাকে সিস্টেম সংক্রান্ত কিছু যা দৈনন্দিন কাজে দরকার পড়েনা একজন ব্যবহারকারীর। এই প্রথম অপশানটা, '`-a`', দিলে `ls` সেই ফাইলগুলোকেও দেখায়। চার আর আট নম্বর লাইনের অপশানদুটো তো আমরা এইমাত্র ব্যবহার করলাম। পাঁচ নম্বর লাইনের অপশানটা একটা ফাইলের ইনডেক্স নোড বা আইনোড দেখায়, সেটার কথায় আমরা আসছি একটু বাদে। ছ নম্বর লাইনের '`-l`' অপশানটা লম্বা লাইনে বড় করে ফাইল সংক্রান্ত অনেকটা তথ্যকে হাজির করে, খুবই কাজে লাগে ফাইলের মালিকানা এবং ব্যবহারযোগ্যতা বুঝতে। পরে দেখব আমরা। '`-R`' বলে, যে ডিরেক্টরিতে বসে `ls` মেরেছি আমরা সেই ডিরেক্টরির মধ্যে যদি কোনো সাবডিরেক্টরি থাকে, তার মধ্যে আরো আরো, সেই ডিরেক্টরিগুলোকেও একই ভাবে দেখিয়ে যেতে। '`-t`' ফাইলগুলোকে সাজায় অক্ষরের অনুক্রমে নয়, তাদের গায়ের সময়চিহ্নের নিরিখে। '`-1`' বলে প্রতিটি ফাইলের জন্যে একটা করে লাইন বরাদ্দ করতে।
আর '`--help`' হল সেই অপশান যা দিয়ে '`ls --help`' কমান্ডটা মেরে আমি এই টেক্সটাকে স্ক্রিনে পেয়েছিলাম। তারপর '`ls --help > ls.help.text`' মেরে একটা ফাইল বানিয়েছিলাম যার নাম ls.help.text । এই ভাবে ফাইলে পাঠানোটাকে কী বলে — মনে আছে আপনার? সেই ফাইল থেকে এখানে তুলে দিলাম। যদিও, অসভ্য ভাবে ছাঁটকাঁট করে, ঠিক যতটুকু আমার মনে হয়েছে আমাদের আলোচনার জন্যে এই মুহূর্তে প্রয়োজনীয় শুধু সেটুকুই রেখেছি। কতটা ছাঁটা বলুন তো? ৯৭ লাইনকে ১১ লাইন বানানো। গোটাটা পড়ার ইচ্ছে হলে, কোনো ব্যাপার না, গ্নু-লিনাক্স সিস্টেম আপনার জন্যে অবারিতদ্বার, বলা মাত্র বানিয়ে দেবে।
আর বলতেও হবেনা, বানানোই আছে। এই হেল্প ফাইলটা যে ফাইলের একটা কুচো। ঠিক পাসওয়ার্ড নিয়ে যেটা করেছিলাম পাঁচ নম্বর দিনে — মনে আছে? '`man 5 passwd`' বলে একটা কমান্ড ব্যবহার করেছিলাম। এই 'man' হল ম্যানুয়াল। '`ls`' কমান্ডটা বিষয়েও সবচেয়ে ভালোভাবে জানার উপায় হল '`man ls`'। শুধু '`ls`' কেন, যেকোনো কমান্ডের বেলাতেই তাই। কিন্তু এই '`man`' ব্যবহার করতে গেলে এই কমান্ডটাকে তো বুঝতে হবে। কেন, তার জন্যে আছে ম্যানুয়ালের ম্যানুয়াল। '`man man`' কমান্ড দিয়ে আমরা 'man' সম্পর্কে জানতে পারব। আর ওই '`--help`' পছন্দটা তো আছেই। ওটা জুড়ে দিয়ে যে কোনো কমান্ড দেওয়া মানেই তার একটা সংক্ষিপ্ত হেল্প ফাইল পেয়ে যাওয়া, যারই বড় ভাই হল ম্যান বা ম্যানুয়াল। এই গোটা ম্যানুয়ালের পাতার সমগ্রতাটা ভরা থাকে গ্নু-লিনাক্স সিস্টেমের ভিতরেই। আপনার অঙ্গুলি-পীড়নের অপেক্ষায়। কিবোর্ডে

টাইপ করে কমান্ড দিলেই হল। এই ম্যান বা ম্যানুয়াল পাতাগুলোর আবার কয়েকটা সেকশন আছে। সেই সেকশন অনুযায়ী আবার খোঁজা যেতে পারে বিশেষ বিশেষ কমান্ডকে। এর সবটুকু এই মুহূর্তে আপনার কাছে স্পষ্ট হবেনা, হতে থাকবে যত আপনি আপনার সিস্টেমটা ব্যবহার করবেন।

এই সেকশনগুলো হল ১, এক্সিকিউটেবল বা চালানো যায় এমন প্রোগ্রাম বা শেল কমান্ড। ২, সিস্টেম কল বা কারনেলের ফাংশন। ৩, লাইব্রেরি কল বা সিস্টেম লাইব্রেরিগুলোর কোনো ফাংশন। ৪, বিশেষ ধরনের কোনো ফাইল — /dev ডিরেক্টরিতে যেমন পাওয়া যায়। ৫, ফাইলের আকার আকৃতি এবং লেখার নিয়ম — /etc/passwd ফাইলে যেমন। ৬, গেমস। ৭, বিভিন্ন ম্যাক্রো প্যাকেজ এবং তার নিয়ম — যেমন man(7) বা groff(7)। ৮, সিস্টেম অ্যাডমিন্সট্রেশন কমান্ড — যাদের শুধু রুট ব্যবহার করতে পারে। আর ৯, বিশেষ ধরনের কারনেল রুটিন। এই সেকশনগুলো পড়ে যদি আপনার এরকম মনে হয়, এগুলো খায় না মাথায় দেয়, নো ফিয়ার, আমি যখন লিখতে পারছি, খায় না মাথায় দেয় এর চেয়ে খুব বেশি কিছু না-বুঝেই, আপনি তখন আলবাত এটা পড়তে পারেন ওটুকু বাদ দিয়েই। তৃতীয় বিশ্বের নাগরিক, আমাদের মস্তিষ্কে ডবল খাঁজ, সাহেবের বাচ্চাগুলো যখন একটা ভাষায় সব কিছু করতে পারে, আমাদের তখন গোটা তিনেক ভাষা তো জানতেই হয়, কী জানব আর কী জানব না এটা জানতে গিয়েই। তাই একসঙ্গে সবটা বুঝে যাবেন নিখুঁত ভাবে এটাই একটা সাহেবী কুসংস্কার। যখন দেখবেন, জানেন কি জানেন না এটা একটু ঘোলাটে, কিন্তু মোটামুটি একটা আন্দাজ পেয়ে গেছেন, কী কাজ বা কী ভাবে করতে হবে, তখনই বুঝতে হবে বুঝে গেছেন। শুধু একটা কথা বলুন তো, এর ৮ নম্বরটা, সিস্টেম অ্যাডমিন্সট্রেশন কমান্ড, যাদের শুধু রুট ব্যবহার করতে পারে, সেই কমান্ডগুলোও তো এক একটা প্রোগ্রাম, এক একটা এক্সিকিউটেবল বাইনারি ফাইল, যাদের নাম লিখে এন্টার মারলে চলে, সেই ফাইলগুলো কোন বা কোন কোন ডাইরেক্টরিতে আছে? পাঁচ নম্বর দিনের ১১ নম্বর সেকশনে, রুটের এর বা পথনির্দেশটা দেখে দেখুন তো বলতে পারেন কিনা? না পারলে, মাইরি বলছি, আপনার জন্যে লিখতে আমার বেশ বোর লাগছে। যাকগে, বেনিফিট অফ ডাউট পেয়ে গেলেন, এতো আর জিএলটি ইশকুলের ক্লাস নয়, যার নোট হিশেবে এই বইটা লেখার শুরু, যে সামনেই আছেন। ধরে নিলাম বলতে পেরেছেন। ২ নম্বর সেকশন মানে সিস্টেম কল নিয়ে কিছু আলোচনা আছে এক আর দুই নম্বর দিনে। আর তিন মানে লাইব্রেরি নিয়ে আলোচনা ছিল তিন নম্বর দিনে, আডার কম্পিউটার ভাষা নিয়ে ভাবনাচিন্তার প্রসঙ্গে। দেখে নিতে পারেন।

সিস্টেমের মধ্যে আপনার সাহায্যার্থে শুধু যে এই ম্যান ভরা আছে তাই নয়, আছে আরো একটা জিনিষ, তার নাম ইনফো —

```
info --help
Usage: info [OPTION]... [MENU-ITEM...]
Read documentation in Info format.
Options:
      --apropos=STRING         look up STRING in all indices of all manuals.
  -d, --directory=DIR          add DIR to INFOPATH.
      --dribble=FILENAME       remember user keystrokes in FILENAME.
  -f, --file=FILENAME          specify Info file to visit.
  -h, --help                   display this help and exit.
      --index-search=STRING    go to node pointed by index entry STRING.
  -n, --node=NODENAME          specify nodes in first visited Info file.
  -o, --output=FILENAME        output selected nodes to FILENAME.

Examples:
  info                       show top-level dir menu
  info emacs                 start at emacs node from top-level dir
  info emacs buffers         start at buffers node within emacs manual
  info --show-options emacs  start at node with emacs' command line options
  info -f ./foo.info         show file ./foo.info, not searching dir
```

`info`। এখানে দেখুন, আমরা সেই ইনফোর হেল্প ফাইল, মানে ‘`info --help > info.help.text`’ মেরে তৈরি, তার থেকে কয়েকলাইন তুলে দিচ্ছি। এবার আর ব্যাখ্যা করছি না, দেখুন তো, আপনি নিজেই পারেন কিনা। ম্যান কমান্ডের খুঁটিনাটি এবার নিজেই পড়ে নিন, ‘`man man`’ মেরে। দু-একটা খুব কাজের কথা বলে দিই। ম্যান দিয়ে যেকোনো কমান্ডের ম্যানুয়াল পেজ খুলে পড়বেন যখন স্পেসবার টিপলে এগোবেন, ‘বি’, B, সুইচটা টিপলে পিছোবেন, আর ‘কিউ’, Q, সুইচটা টিপলে বেরিয়ে যাবেন। এটা অভিমন্যুকে দেখে শিখুন, কোথাও ঢোকার আগে মনে রাখবেন বেরোনোর উপায়। কোনো কিছু খুঁজতে হলে লাগবে ‘স্ল্যাশ’ বা / সুইচটা। এটা টেপা মাত্র দেখবেন একদম নিচে এটা ফুটে উঠেছে, সেখানে টাইপ করে দিন আপনি যা খুঁজতে চান, এবং এন্টার মারুন। এই চারটে সুইচ শুধু এই ম্যান না, আরো বহু গ্নু-লিনাক্স কমান্ডের বেলাতেই সত্যি। এবার ধরুন, আপনি যা খুঁজতে চাইছেন সেটা কোনো কমান্ড নয়, একটা কোনো বিশেষ শব্দ, সেটা কোথায় কোথায় আছে, কী

আছে, এটা আপনি জানতে চাইছেন। ধরুন কথাটা হল CD, সিডি নিয়ে আপনার সিস্টেমের ম্যানুয়াল পেজগুলোয় কী বলা আছে আপনি জানতে চাইলেন। আপনি কমান্ড দিন '`man -k CD`', দেখুন স্ক্রিনে একটা তালিকা ফুটে উঠেছে। গোটা তালিকাটার প্রত্যেক লাইনেই বাঁ দিকে একটা শব্দ, তারপর ব্রাকেটে একটা নম্বর, তারপর ডানদিকে লেখা বাঁদিকের ওই শব্দটা কী তাই নিয়ে একটা বিবৃতি। যেমন আমার সুজে ৮.২ সিস্টেমে আমি '`man -k CD`' করে যে তালিকাটা পেলাম সেটার একটা লাইন —

```
cdrecord (1)         - record audio or data Compact Discs from a master
```

একদম বাঁদিকে সিডিরেকর্ড বলে একটা কমান্ডের মানে একটা এক্সিকিউটেবল প্রোগ্রামের নাম, তারপর ব্রাকেটে লেখা (১), কারণ, একটু আগের ম্যানুয়াল পেজগুলোর সেকশনের তালিকাটা দেখুন, ১ নম্বর সেকশনে থাকার কথা এক্সিকিউটেবল প্রোগ্রাম বা শেল কমান্ডগুলোর। এবং ডানদিকে লেখা সিডিরেকর্ড সম্পর্কে একটা কুচো মন্তব্য। সিডিরেকর্ড বস্তুটা ঠিক কী করে। এবার আপনি যদি এই সিডিরেকর্ড সম্পর্কে জানতে চান তাহলে কমান্ড দেবেন '`man 1 cdrecord`'। এবার বলতে পারেন ওই কমান্ডটায়, `man 5 passwd`, আমরা ১ না দিয়ে ৫ দিয়েছিলাম কেন? সেকশনের তালিকার সঙ্গে মিলিয়ে দেখুন।

এখানে একটা ঘাপলা আছে। ঘাপলাটাকে একদম শাঁসেজলে ফিল করতে চাইলে এই কমান্ডটা টাইপ করুন, '`man -k X`', এবার আপনার প্রসেসর যদি ওইসব চৌষট্টি বিট ফিটের মালেদের একজন না হয়, আপনি স্বচ্ছন্দে একটা সিগারেট ধরান। কারণ, বেশ কিছু সময় ধরে এক স্ক্রিন আপনি এক পলকের বেশি দেখতে পাবেন না। পাতার পর পাতা চলতেই থাকবে। এই '`X`' মানে মূলত এক্স-উইনডোজ বা গ্রাফিকাল ইউজার ইন্টারফেস বা গুই নিয়ে আলোচনা আছে আমার সুজে সিস্টেমে মোট ২৮২৯ খানা ম্যানুয়াল নিবন্ধে। সিস্টেম ম্যানুয়ালের মোট সাইজ বুঝতে পারছেন? একটু আগে '`CD`' খুঁজেছিলাম যখন তখনো তার সংখ্যা পেয়েছিলাম ৩৮ খানা, তার মানে এক স্ক্রিনের মানে ২৫ লাইনের বেশি। তাহলে খুঁজবেন কী করে? শুধু ডিসপ্লেটা যখন থেমে যাবে — মানে তালিকার শেষ পাতায়? এটা একটা স্ট্যান্ডার্ড কথা হল? লেস আছে কী করতে? লেস নিয়ে একটা মজার কথা চলে, লেস ইজ মোর দ্যান মোর। মানে, মোর বলে একটা পেজার আগেই ছিল, সেই পুরোনো মোর-এর সঙ্গে আরো কিছু কাজ এনে তৈরি করা হয় এই লেস বলে পেজারটাকে। পেজার বলতে একটা সফটওয়ার যা কোনো টেক্সটকে পাতা বাই পাতা পরপর দেখিয়ে যায়। এবার আর স্ক্রিন একটি নদীর নাম, স্বপ্নের ছন্দে সে বহিয়া যায় — তা নয়, তাকে বেঁধে ফেলা হয়েছে লেস-এর শেকলে। লেস-এর পোষাকের অনির্বচনীয়তা নিয়ে আপনার চোখে হয়তো ধরা দেবে না, কিন্তু কাজ ভারি ভালো চলে যাবে।

`man -k CD | less` — এটায় আমরা মূল কমান্ডের আউটপুট বা ফলাফলটাকে পাইপ করে দিলাম লেস দিয়ে। এবার ধীরে সুস্থে পাতাটাকে নামান। ঠিক একটু আগে ম্যান পেজের বেলায় যে চারটে সুইচ বলেছিলাম, স্পেসবার, বি, কিউ এবং স্ল্যাশ — সেই চারটেই একই কাজ করে এখানেও। শুধু এখানে না, ইনফো-তেও করবে, আরো অনেক কমান্ডেও। এবার বলুন তো, মোট কত লাইন ছিল ম্যান-এর ফলাফলে, ২৮২৯, যে বললাম, বললাম কী করে? গুণে গুণে? প্লিজ, আমার বাবার দিককার কাস্ট কায়স্থ, মানে ক্লার্ক, কিন্তু অতো মাছি মারার প্র্যাকটিস আমার কোনোদিনই নেই।

এবার দেখুন, লেস করেও যদি খুঁজতে চান, বহু বহু পাতা অনেক ক্ষেত্রেই, তখন খুঁজে পাওয়াটা অসম্ভব আপনি যা খুঁজছেন। তখন অন্য একটা তরকিব কাজে লাগান। তার নাম গ্রেপ। ধরুন, আপনি সে সিডি নিয়ে খুঁজছেন সেটা আসলে আপনার অডিয়ো সিডি নিয়ে কোনো কৌতূহলে। তাহলে এবার ম্যান-এর আউটপুটটাকে পাইপ করে দিন গ্রেপ-এ।

`man -k CD | grep 'audio'` — এর থেকে স্ক্রিনে যা পাবেন সেটা এই পাতার চওড়ায় পাশাপাশি ধরানোর জন্যে একটু কেটেছেঁটে অনেকটা এইরকম —

```
cdda2wav (1)     - a sampling utility that dumps CD audio data into wav files
cdrecord (1)     - record audio or data Compact Discs from a master
cdparanoia (1)   - an audio CD reading utility with extra data verification
cdrdao (1)       - writes audio CD-Rs in disc-at-once mode
cda (1)          - Compact disc digital audio player utility
xmcd (1)         - CD digital audio player utility for X11/Motif
```

এবার ধরুন, এইটা যখন স্ক্রিনে দেখছেন, হঠাৎ খেয়াল করলেন, আরে, চারপাশটা এত নিস্তব্ধ লাগছে কেন, স্ক্রিনে '`date`' মেরে বা ঘড়িতে দেখলেন, আরে আড়াইটে বেজে গেছে, ধুর ছাতা, এত রাতে অডিয়ো সিডির এত প্রোগ্রাম খুঁজে পেয়েই বা কী হল, বাজানো তো যাবেনা, পাড়ার আগেও বাড়ি, তারপর কালকের অফিস, একটা দীর্ঘশ্বাস ফেলে আপনি এই লাইনকটা তুলে রাখতে চাইলেন একটা ফাইলে, কালকে যাতে এত খোঁজাখুঁজি করতে না হয়, আপনি কমান্ড দিলেন

`man -k CD | grep 'audio'> audio.cd.program.text`, তারপর একবার শেষ দেখে নিতে চাইলেন, ফাইলটা ঠিক সেভ হয়েছে কিনা, বা তার সাইজ ঠিক কতটা। '`ls -sh audio.`' টাইপ করে দিলেন স্ক্রিনে, কিন্তু সাবধান, এবার কিন্তু এখনো আপনি এন্টার সুইচটা মারেননি। তার জায়গায় মারবেন ট্যাব সুইচটা। এটা ব্যাশ শেলের কমান্ড কমপ্লিশন

বা নিজে থেকে আদেশের শূন্যস্থান পুরন করে দেওয়ার কাজ। যদি আপনার ওই ডাইরেক্টরিতে আর কোনো ফাইল না থাকে যার নাম ‘audio.’ দিয়ে শুরু, তাহলে ইতিমধ্যেই আপনার স্ক্রিনে জ্বলজ্বল করছে আপনার গোটা দিতে চাওয়া কমান্ডটা, ‘ls -sh audio.cd.program.text’। আর যদি একাধিক ফাইল থাকে যারা প্রত্যেকেই ‘audio.’ দিয়ে শুরু, তাহলে স্ক্রিনে পাবেন আপনার কমান্ডটা যতটা আপনি টাইপ করেছেন, তার নিচে ওই কটা ফাইলের নাম। যে অব্দি গেলে এই নামটা সম্পূর্ণ নিজস্ব চেহারা পেয়ে যাবে সেই অব্দি আপনি টাইপ করে দেবেন, তারপর গোটাটুকু ভরে দেবে ব্যাশ নিজেই। যেদিন আপনার একটা দুটো বর্ণ টাইপ করার পরই আপনার বাঁ হাতের কড়ে আঙুল ট্যাব সুইচের দিকে যেতে শুরু করে দেবে, সেদিন বুঝবেন আপনি ব্যাশ শেলে অভ্যস্ত হতে শুরু করেছেন। ধরুন, আপনি ‘ls -sh audio.’ টাইপ করার পর দেখলেন স্ক্রিনে ফুটে উঠেছে আপনার টাইপ করা কমান্ডটুকু এবং নিচে দুটো ফাইলের নাম, ‘audio.cd.program.text’ এবং ‘audio.filetypes.list’, এবার আপনি জাস্ট একবার ‘সি’ সুইচটা টিপে ট্যাব মারলেই ব্যাশ গোটা কমান্ডটা ফুটিয়ে তুলবে, কারণ, ‘সি’-র থেকে দুটো ফাইল আলাদা, ওই ‘সি’-টা আছে শুধু আপনার দেখতে চাওয়া ফাইলটাতেই। এবার যেই এন্টার মারবেন, স্ক্রিনে ফুটে উঠবে ফাইলটার নাম এবং মানববোধ্য সাইজ, আগেই তো বলেছি। সাইজটা দেখার পর নিজের কৃতিত্বে মুগ্ধ আপনার মনে হল, আরে পরের বার এতোটা টাইপ যাতে না করতে হয়, ব্যাশের কমান্ড কমপ্লিশন ফাংশনটার পুরো মজাটা যাতে পাওয়া যায়, তার জন্যে ‘audio.filetypes.list’ ফাইলটার নামটা তো বদলে দিলেই হয়। আপনি অন্য একটা নামে, ‘sound.filetypes.list’, ফাইলটা কপি করবেন। কপি করার কমান্ড ‘cp’ তো আমরা আগে থেকেই জানি, পাঁচ নম্বর দিনে তিন নম্বর সেকশনে ছিল।

কমান্ড দিলেন, ‘cp audio.filetypes.list sound.filetypes.list’। এবার এন্টার মারলেন, কমান্ড প্রম্পট সঙ্গে সঙ্গে চলে এল, মানে কাজটা হয়ে গেছে। কাজটা না হলে জানাতে পারে, চুপচাপ ফিরে এল মানে কাজ সমাধা, শেল এভাবেই কাজ করে। এবার একবার ‘ls’ মেরে দেখে নিলেন। দেখতে গিয়েই মনে হল, ও হরি, পুরোনো ‘audio.filetypes.list’ ফাইলটাও তো রয়ে গেল। কোই পরোয়া নেই ফাইল ওড়ানোর কমান্ড ‘rm’ — সেটাও তো আপনি জানেন। সহজেই উড়িয়ে দিলেন ফাইলটা। ‘rm audio.filetypes.list’। এন্টার মারলেন। প্রম্পট ফেরত এল। মানে ফাইল উড়ে গেছে। একবার ‘ls’ মেরে দেখে নিলেন। তারপর মাথায় এল, আরে প্রথমেই কপি করে, তারপর আরএম বা রিমুভ না করে, একবারেই তো করা যেত ‘mv’ দিয়ে, সেটা তো ফাইলকে নড়ায় বা নাম বদলায়। তখন একটা কমান্ডেই হয়ে যেত — ‘mv audio.filetypes.list sound.filetypes.list’। যাকগে।

শুতে যাওয়ার আগে এখনো একটা কাজ বাকি আছে আপনার। আপনি একবার দেখে নিতে চাইলেন, ফাইলটা ঠিক কোথায় আছে, আপনার হোম ডিরেক্টরির কোথায়? কোন সাবডিরেক্টরিতে? এর জন্যে আপনি যে কমান্ডটা ব্যবহার করবেন সেটা হল ‘pwd’। মানে, কোন ডিরেক্টরিতে কাজ করছি সেটা দেখাও। প্রিন্ট ওয়ার্কিং ডিরেক্টরি। এর পরে আপনি যে ডিরেক্টরিতে সেভ করেছেন ফাইলটা, তার ঠিকানাটা ফুটে উঠবে। ধরুন এটা আমি করছি। আমি ফাইলটা সেভ করেছি গার্বেজ বলে একটা ডিরেক্টরিতে। এই গার্বেজ ডিরেক্টরিটা, ধরুন অডিয়ো নামের ডিরেক্টরিতে। যে অডিয়ো ডিরেক্টরিটা আছে আমার হোম ডিরেক্টরির মাল্টিমিডিয়া বলে একটা ডিরেক্টরিতে। এই অবস্থায় আমি ‘pwd’ মারলে স্ক্রিনে ফুটে উঠবে —

/home/dd/multimedia/audio — এটাই এই মুহূর্তে আমার কাজের ডিরেক্টরি। কাল এসে আর খুঁজে পেতে এবং অডিয়ো সিডির গান বাজাতে অসুবিধে হবেনা।

পাঁচ নম্বর দিন আর আজকের শুরু থেকে আমরা যে ষোলটা কমান্ড এখনো পেয়েছি সেগুলো হল — mkdir, rmdir, cd, rm, mv, cp, ls, who, tar, less, wc, man, bzip2, info, grep, pwd। এর মধ্যে আবার mkdir, rmdir, cd কমান্ডগুলো আলোচনায় আনতে পারিনি কারণ এখনো আমরা রিলেটিভ এবং অ্যাবসলিউট পাথ, আপেক্ষিক এবং চূড়ান্ত পথনির্দেশ বলতে কী বোঝায় সেই প্রসঙ্গ এখনো আনতে পারিনি। mv কমান্ডটাকে নড়ানোর কাজে কী ভাবে ব্যবহার করা যায় সেটাও বলা হয়নি সেই একই কারণে। সেই পাথ বা পথ-এর কথা বলতে গেলে একটু ফাইল সিস্টেম জানতেই হবে। এবার আসবে। প্রথমে ফাইল, তারপর ডাইরেক্টরি সিস্টেম, তারপর ফাইল সিস্টেম। কিন্তু যা বলেছিলাম, এই লেখাটায় আসলে আমরা দুজনেই খেলছি, নইলে খেলাটাই হবেনা। আপনি এখন যতদূর সম্ভব তার চেয়ে একটু বেশি পরিশ্রম করে এই কমান্ডগুলোয়, এদের ম্যান আর ইনফোয় অভ্যস্ত হয়ে নেবেন আপনার সিস্টেমে। যত পারা যায় নানা আলাদা আলাদা অপশান কমান্ডের সঙ্গে লাগিয়ে লাগিয়ে দেখবেন তাতে কী হয় কী না হয়। আর যতটা মারেন ম্যানুয়াল পড়া অভ্যেশ করে ফেলুন এখন থেকে। সামনে দুঃখের দিন আসছে। এখানে যত বেশি পরিশ্রম করবেন, তত আপনার পরে সুবিধে হবে। অনেকটা গানের রিওয়াজের মত। টাচ-টাইপিং বা দশ আঙুলে টাইপ করার মত। এই টাচ টাইপিং-এর জন্যে একটা চমৎকার গ্নু সফটওয়ার আছে, জিটাইপিস্ট। পরে আমরা আসব সেসবে। এখানে হঠাৎ একটা কথা মাথায় এলো, বারবার আ মি লিখে যাচ্ছি পেজ আপ পেজ ডাউন করে পড়ার কথা, তার মানে আমি কি ধরে নিচ্ছি যে আপনি এটা কম্পিউটারে পড়ছেন? যদি তাই হয় তাহলে সেটা কী? উইনডোজ একটা প্রতিবেশ? যদি তাই হয়, তাহলে তো আপনি এটা পড়াকালীন করে দেখতে পারবেন না। যদি এটা গ্নু-লিনাক্স প্রতিবেশ হয়,

তাহলে সেটা একই সঙ্গে একটা এক্স-উইনডোজ-ও। গুহনোম হোক, কেডিই হোক, ব্ল্যাকবক্স বা ফ্লাক্সবক্স বা অন্যকিছু যাই হোক। কারণ এটা একটা পিডিএফ ফাইল, যা প্রিন্ট করা যাবে কমান্ড লাইনে, কিন্তু কমান্ড মোডে পড়ার কোনো প্রকৌশল আছে বলে জানিনা। কার্সেস দিয়ে ফ্রেমবাফার দিয়ে কিছু কখনো তো শুনিওনি। এফবিআই দিয়ে যেমন গ্রাফিক্স দেখা যায়, বা এমপ্লেয়ার যেমন কমান্ড মোডেই দিব্য সিনেমা দেখায়। তার মানে আপনার মেশিনে গ্নু-লিনাক্সে এক্স-উইনডোজ চলছে। এই অবস্থায় আপনি একটা টার্মিনাল খুলে নিন। এবং সেখানে কমান্ডগুলো পড়তে এবং করতে থাকুন। আর সঙ্কর্ষণরা যেমন প্রাণপণ দৌড়চ্ছে এটাকে হার্ড প্রিন্টে মানে বই আকারে বার না-করে ছাড়বে না, সেটা যদি শেষ অব্দি ঘটে যায়, যদি, কারণ, একটা বিরাট পরিমাণ টাকা লাগবে বইটা বার করতে, সেটা কলকাতা লাগ-এর বা এফএসএফ-ওয়েস্টবেঙ্গলের নেই — বই আকারে যদি পড়েন তাহলে অবশ্য সমস্যা নেই, কমান্ড মোডেই করে দেখতে পারবেন বই পড়ার পাশাপাশি।

এখানে একটা প্রাক্টিকাল সমস্যার কথা বলি, যে জিএলটি ক্লাসের সূত্রে প্রথমে ক্লাসনোটস এবং তারপর ক্রমে গতরবৃদ্ধি হতে হতে ক্রমে এই বইয়ের আকারে চলে এল, তার ক্লাস চলাকালীন অশেষ সমস্যাটার কথা বলেছিল। ও তখনো কেবল সিস্টেমে ঢুকতে শুরু করেছে, তখনো চেনাজানা তৈরি হয়নি, ওর নাকি এটা খুব ধাঁধার মত লাগছ, কাদের ম্যানুয়াল পেজ পড়া দিয়ে শুরু করব? মানে কমান্ডগুলোর বিবরণ নয় ম্যানুয়াল পেজে লেখা আছে, যেমন ls কমান্ডটাকে জানলাম '`man ls`' কমান্ড দিয়ে কিন্তু এই কমান্ডটার কথাই বা জানব কী করে? মানে কাকে ম্যান দিয়ে খুঁজব? আপাতত এটুকুই বোধহয় জেনে রাখা ভালো, /bin ডিরেক্টরিতে যে যে ফাইল আছে, তাদের ম্যানুয়াল পেজ পড়া দিয়েই শুরু করা ভালো। এই /bin ডিরেক্টরিতে যারা বিরাজ করেন তারা সবাই একজিকিউটেবল মানে প্রোগ্রাম বা বাইনারি ফাইল। এর হাল হকিকত আমরা বেশ জেনে যাব আজকের আলোচনা শেষ হওয়ার আগেই। এই ডিরেক্টরির প্রোগ্রামগুলো একের পর এক ধরে ধরে ম্যান পেজ খোলা, সব সময়েই কি লোকে পড়ে নাকি, ছি, গানবাজনাসিনেমা আছে কী জন্যে, একটু নাড়াচাড়া করা, সিস্টেমের সঙ্গে ঘনিষ্ঠতা বাড়ানোর একটা ভালো রাস্তা। কারা কারা আছেন এই ডিরেক্টরিতে জানব কী করে? কেন, সিস্টেমের যে কোনো ডিরেক্টরিতে দাঁড়িয়েই '`ls /bin`' মারুন। দরকার হলে রিডাইরেক্ট করে নিন একটা টেক্সট ফাইলে, '`ls /bin > binary.file.list`' বা যে কোনো নামের একটা ফাইলে। একটা কথা মাথায় রাখুন, এই সমস্যাটাও হতে দেখেছি জিএলটি ইশকুলে, যাকে তাকে কিন্তু ক্যাট করা বা রিডাইরেক্ট করে ফাইল বানানো যায়না, মানে যায় তো সবই, একটা বাইনারি ফাইলকে যদি আপনি ক্যাট করেন বা রিডাইরেক্ট করে একটা ফাইল বানান, সেটাকে লেস দিয়ে পড়ার চেষ্টা করেন, শূন্য নম্বর দিনের ভিশুয়াল চ্যাংড়ামির গোত্রের জিনিষপত্তর পাবেন। কিন্তু ওই বাইনারি ফাইলের ম্যানুয়ালটাকে রিডাইরেক্ট করে ফাইল বানালে বেশ পড়তে পারবেন।

২.১।। ফাইল

ফাইল প্রাথমিকভাবেই একটা বিমূর্ত জিনিষ। এই গ্নু-লিনাক্স ইশকুল পাঠমালার এক নম্বর দিনে আমরা একটা সাঙ্কেতিক বা সিম্বলিক ব্যবস্থার কথা আলোচনা করেছিলাম, সেই বিমূর্ত একটা চিহ্ন-কাঠামো, যা দিয়ে আমি কম্পিউটারকে বুঝছি। এই যে হার্ড ডিস্ক বুঝছি /dev/hda বা /dev/hdb দিয়ে, ফ্লপি ডিস্ক বুঝছি /dev/floppy দিয়ে, সিডিরম বুঝছি /dev/cdrom দিয়ে — এগুলো কী? এক একটা নাম, চিহ্ন, এর পিছনে একটা ভৌত বাস্তবতা রয়েছে, যে বাস্তবতাটাকে মাথায় রাখছে কারনেল তথা অপারেটিং সিস্টেম। এই সিম্বলাইজেশন বা চিহ্নীকরণ প্রক্রিয়াটা ঘটতে শুরু করে যেখান থেকে তার নাম ফাইল। ফাইল, তার নিজের নামের আশ্রয়ে বিমূর্ত করে তোলে কিছু বাইট সমাহারকে যাদের ভৌত বাস্তবতাটাকে আমরা তখন আর ভাবছি না, আমার কাছে তখন ফাইল মানেই ডিস্কের গায়ে বরাদ্দ করা ওই বাইট সমাহারে লিখিত তথ্যটা। এতে ওই ঘসঘস করে ঘুরতে থাকা চৌম্বক প্রলেপ লাগানো গোলাকৃতি প্ল্যাটারগুলোকে, তাদের ভীষণ কাছ দিয়ে পরিক্রমণশীল হেডগুলোকে, তাদের নড়াচড়াকে, চৌম্বক তলের উপর ওই হেড দিয়ে তথ্য লেখা বা পড়ার কায়দাকে, ভুলে থাকছি আমি। শূন্য নম্বর দিনে এই নিয়ে আলোচনাটা মনে করুন। যে বাস্তবতাটাকে ফাইল গোপন করে রাখে তার নামের পিছনে। এখন থেকে ওই বাইট সমাহার পরিচিত হবে ওই নাম দিয়ে। মাথায় রাখুন, তথ্য মানে তথ্যহীনতাও হতে পারে। ব্ল্যাংক বা ফাঁকা ফাইল যেখানে কোনো তথ্য লেখা নেই।

প্রথমে এই ফাইল নামকরণের প্রক্রিয়াটাকে বোঝার চেষ্টা করা যাক। ফাইলের নাম শুধু যে আপনি দেন তা নয়, আপনি যে প্রোগ্রামগুলো চালাবেন তারা বহু ফাইল তৈরি করে, অপারেটিং সিস্টেম তার কাজ করার প্রক্রিয়ায় বহু ফাইল বানায়। এই ফাইল বানানো মানেই তো নির্দিষ্ট কিছু ভৌত বাইট সমাহারে গেঁথে রাখা তথ্য সমাহারের জন্য আলাদা একটা পরিচিতি তৈরি করা, আলাদা একটা নাম দিয়ে। আবার অজস্র ফাইল থাকে যারা একটা প্রসেস চলাকালীন তৈরি হয়, প্রসেস মরে গেলে তারাও নিশ্চিহ্ন হয়ে যায়, তাদের আলাদা করে আর সেভ করা হয়না ডিস্কে। প্রসেসটা যতক্ষণ চলে ততক্ষণই তারা বেঁচে থাকে, র‍্যামে বা ভার্চুয়াল মেমরিতে, সোয়াপ ফাইলে, মানে হার্ড ডিস্কে আলাদা করে রাখা অস্থায়ী ফাইল অস্থায়ীভাবে লেখার জায়গায়। এটা খুব ভালো বোঝা যায় কোনো কিছু লেখার সময়। লিখছি, নতুন শব্দগুলো জমা হচ্ছে, জমা হচ্ছে কোথায়? ফাইলটার স্থায়ী শরীরে নয়, কারণ, এখুনি যদি আপনি বেরোতে চান, আপনাকে সিস্টেম জিগেশ করবে, ফাইলটা কি সে সেভ

করবে? তার মানে এখনো তাকে সেভ করা হয়নি। হঠাৎ কারেন্ট চলে গেলে আপনি পরে আর এই না-সেভ করা অংশটা পাবেন না। এই অংশটা অস্থায়ী ভাবে লেখা থাকে, সেভ করার সময়ে তাদের স্থায়ী ফাইলের অংশ করে দেওয়া হয়। হোক অস্থায়ী, তাও তো সেটা একটা লেখা, ওই সময়টুকুর জন্যে তাকে সেইখানে একটা বিশেষ পরিচিতিতে থাকতে হচ্ছে। ধরুন আপনি একই সঙ্গে দুটো ফাইল এডিট করছেন, সেখানে একটার অংশ তো অন্যটায় সেভ হচ্ছেনা, তার মানে ওই অস্থায়ী সময়টুকুর জন্যেই তার একটা পরিচিতি আছে — অস্থায়ী একটা নামও আছে, সিস্টেম যে অস্থায়ী নামের ফাইলে তথ্য লিখছে বা পড়ছে। আপনাকে জানতে হচ্ছে না, কিন্তু নাম একটা আছেই।

২.২।। ফাইলনাম এবং এক্সটেনশন

এই নাম কী ভাবে দেওয়া হবে তার খুঁটিনাটি নিয়ম আবার সিস্টেম থেকে সিস্টেমে আলাদা। ফাইলনামে চিহ্নের সংখ্যা সিস্টেম থেকে সিস্টেমে ৮ থেকে ২৫৫ অব্দি হয়। এবং সংখ্যা বা অক্ষর বা অন্য কিছু চিহ্ন ব্যবহার করা যায়। কোনো কোনো সিস্টেমে বড় হাতের বা ক্যাপিটাল কেস অক্ষর এবং ছোট হাতের বা স্মল কেস অক্ষর আলাদা বলে গণ্য হয়, কোথাও হয়না। যেমন গ্নু-লিনাক্সে dog বা Dog বা dOg বা doG বা DOg বা DoG বা dOG বা DOG প্রত্যেকেই আলাদা ফাইল, কিন্তু এমএসডস বা উইনডোজে এই সব কটা নামই চিহ্নিত করবে একই ফাইলকে, সেটা সে দেখাবে dog বা Dog বা DOG। আগের আটটার যেকোনো একটা নামে একটা ফাইল থাকলেই সেই ডাইরেক্টরিতে আর ওই আটটার অন্য কোনো নামে কোনো ফাইল তৈরি করতে দেবেনা। এমএসডস বা উইনডোজ ফাইলনামে দুটো অংশ থাকে। কাঠামোটা হল *.*। এই কাঠামোয় প্রথম * টা এমএসডসে হত ৮ বা তার কম অক্ষরের, পরের * টা তিন বা তার কম। প্রথম *-টাকে বলে ফাইলনেম, আর পরের *-কে বলে এক্সটেনশন। উইনডোজে একই সঙ্গে প্রতিটা ফাইলনামের দুটো সংস্করণ লিখে রাখা হয় — একটা ওই এমএসডসের ছক মেনে, ৮.৩-এ, অন্যটা হল ২৫৫ চিহ্নের, ওই ছকটা না-মেনে। দুটোই ব্যবহার করা যায়। এমএসডস প্রম্পটে যেমন দেখবেন, দুটো নামই দেখায়, প্রাথমিক ভাবে ডস নামটা, ডানদিকে উইনডোজ নামটা। আপনি যদি উইনডোজে C:\ মানে রুট ডিরেক্টরিতে দাঁড়িয়ে cd My Documents লিখে এন্টার মারেন, ডস প্রম্পট আপনাকে দেখাবে — Too many parameters – Documents, অর্থাৎ সিস্টেম এই নামটার মানেই বুঝতে পারছে না। কিন্তু cd mydocu~1 লিখে এন্টার মারলে পৌঁছে দেবে C:\My Documents\ ডিরেক্টরিতে। কিন্তু উইন্ডোজ এক্সপ্লোরার দিয়ে সিস্টেম ব্রাউজ করার সময়ে দেখাবে গোটা নামটাই। এটা তার দেখানোর নাম, সিস্টেম তাকে বোঝে 'mydocu~1' দিয়ে, যার অক্ষরের সংখ্যা ৮, সেই ডস নিয়ম।

গ্নু-লিনাক্সে সিস্টেমের কাছে এই এক্সটেনশন ব্যাপারটার কোনো মানেই নেই, মানেটা আছে আপনার কাছে, আপনি একই ধরনের অনেকগুলো ফাইলকে একই এক্সটেনশন দিয়ে রাখলে পরে আপনার চিনতে সুবিধে হবে, যে কারণে আমরা প্রবন্ধগুলোর নাম দিচ্ছিলাম essay.1.text, essay.2.text, essay.3.text, ... ইত্যাদি। এখানে দেখুন নামের আসলে দুটো নয় তিনটে খণ্ড। এই নামগুলো সংখ্যা ব্যবহার করেছে, সেখানে শুধু অক্ষর ব্যবহার করা যেতে পারত। আমি যেমন এই নামগুলো দিতাম, 01.essay.text, 02.essay.text, 03.essay.text, ... ইত্যাদি, যাতে আমায় ব্যাশ কমান্ড কমপ্লিশনের কাছে শুধু গোড়ার দুটো অঙ্ক দিয়ে ট্যাব মারলেই কাজ চলে যায়। আর প্রথম ডট বা বিন্দুটা দেওয়ার একটা কারণ এই যে ls মারলে পরপর দেখতে বেশ সুন্দর লাগে। একদম ডানদিকের যে অংশটা '.text' — এর থেকে আপনি, সিস্টেম নয়, আপনি, চেনেন যে এটা একটা টেক্সট ফাইল। আপনি এটা '.txt' দিতে পারতেন, '.essay' দিতে পারতেন, '.probondho' দিতে পারতেন, বা যা আপনার খুশি। এর আবার কিছু প্রথা বা কনভেনশনও গড়ে উঠেছে। যেমন, '*.c' মানে সি প্রোগ্রামের কোড লেখা ফাইল, '*.o' মানে কম্পাইলেশনের সময়ে তৈরি অবজেক্ট ফাইল, '*.html' মানে নেটের হাইপার টেক্সট মার্কআপ ল্যাংগোয়েজের ফাইল, '*.ps' মানে পোস্টস্ক্রিপ্ট ফাইল, '*.mp3' মানে এমপেগথ্রি মানে যার থেকে গান শুনি, '*.avi' মানে যার থেকে সিনেমা দেখি ইত্যাদি। বলুন তো '*.h' বা '*.tar.bz2' বা '*.pdf' কী ধরনের ফাইল? মনে করার চেষ্টা করুন, প্রত্যেকটাই উল্লেখ করা আছে আগের দিনগুলোর লেখায়। আর আন্দাজ করার চেষ্টা করুন তো, '*.conf' কী ধরনের ফাইল? একটা জিনিষ দেখুন, এই যে আমরা এক্সটেনশনগুলোকে উল্লেখ করছি '*.' দিয়ে, এর মানে ওই '*' অংশে যা খুশি আসতে পারে, তার পর একটা বিন্দু বা ডট '.' থাকবে, তারপর আসবে এক্সটেনশনটা। এটা একটা রেগুলার এক্সপ্রেশন, মানে একটা ছক, শেল যা দিয়ে সব গুলো প্রবন্ধের ফাইলকে চিনেছিল পাঁচ নম্বর দিনে।

এবং এই যে বললাম না, 'যা খুশি আসতে পারে' — এটা যে কী ভীষণ বিপজ্জনক, তা টের পাওয়ার জন্যে আপনি হাতে নাতে একটা জিনিষ করতে পারেন। একটা ফাইল তৈরি করুন। ফাইল তৈরি করার দুটো খুব সহজ উপায় হল টাচ এবং ক্যাট। টাচ আসলে অন্য কাজের জন্যে তৈরি, তার একটা ব্যবহার এই ফাইল বানানো। ধরুন কমান্ড দিলেন '`touch essay`'। এর মানে '`essay`' নামে একটা ফাইল তৈরি হয়ে যাবে আপনি যে ডাইরেক্টরিতে আছেন সেখানে। কিন্তু এটা হবে শূন্য বা ব্ল্যাংক ফাইল। এত শূন্য বা এত ব্ল্যাংক যে তাদের মধ্যে খড়ও ভরা নেই, বা ভাঙা কাঁচের উপর ইঁদুরের পায়ের শব্দের মত শুকনো গলায় তারা ফিসফিসও করেনা, শুধু একটা পরিচিতি — একটা ফাইলনাম, এবং তার সঙ্গে একটা টাইমস্ট্যাম্প, মানে কখন বানানো হল, আর আরো দু-চারটে তথ্য।

ওই একই ফাইল ক্যাট দিয়ে বানাতে হলে আমাদের ব্যবহার করতে হত ‘`cat > essay`’। এখন আমরা আর এন্টার মারার কথা উল্লেখ করছিনা, এত বার করেছি, যে সেই ব্যাপারটা আমাদের মাথায় এন্টার করে গেছে বলেই আশা করা যায়। ক্যাট দিয়ে কাজ করতে চাইলে কাজটা কিন্তু এখানেই শেষ নয়। (ক্যাটের কাজটা কী বলুন তো, আগেই বলে এসেছি, মনে করতে পারছেন?) ক্যাট এর চিহ্নটা দেখুন, ও রিডাইরেক্ট করছে, চালান করছে। কিন্তু কাকে? আপনাকে, মানে আপনার টাইপকে, এখন আপনি যা টাইপ করবেন। যা খুশি। দেখবেন, আপনার টাইপ করার জায়গা বানাতে ক্যাট কমান্ড প্রম্পটটাকেই হাওয়া করে দিয়েছে, সেখানে জ্বলজ্বল করছে কার্সর, আপনার টাইপ করা অক্ষর যেখানে ফুটে উঠবে।প্রতি লাইনের শেষে একবার করে এন্টার মারবেন, নতুন লাইন শুরু হবে, তারপরে যখন আপনার মনে হবে, টাইপ করতে করতে আপনি নিজেই প্রায় টাইপরাইটার হয়ে গেছেন, তখন আর কিছু এন্টার করবেন না, একজিট করবেন, কন্ট্রোল-ডি মেরে, দেখবেন এতক্ষণের অদৃশ্য কমান্ড প্রম্পট ফের সগৌরবে ফিরে এসেছে। এবং একবার `ls` মারলে দেখতে পাবেন আপনার বানানো ফাইল। আর তার মধ্যে আপনার যে সূক্তিরত্নাবলী টাইপ করে রেখেছিলেন সেটা দেখতে চান, মারুন ‘`cat essay`’। স্ক্রিনে দেখিয়ে দেবে। এবার আর এটা শূন্য নয়, আপনার টাইপ করা লেখা ভরা রয়েছে।

এবার, খ্যাঁচটা এইখান থেকে শুরু, নামটা ‘`essay`’ না-দিয়ে দিন ‘`-essay`’। মজার কথা কী বলুন তো এই নামে ফাইল আপনি ‘`touch -essay`’ দিয়ে তৈরিই করতে পারবেন না। যদিও, ‘`cat > -essay`’ দিয়ে পারবেন। ‘ls’ সেভাবেই দেখাবে এটাকে যে ভাবে অন্য ফাইলদের দেখায়, আপনি যে নাম দিয়েছেন সেই নামেই। তাহলে? সমস্যাটা কোথায়? এবার, আপনি ‘`rm -essay`’ দিয়ে এটাকে ওড়ানোর, ‘`mv -essay`’ দিয়ে নাম বদলানোর, বা ‘`cp -essay essay`’ দিয়ে এটাকে ‘`essay`’ নামে কপি করার চেষ্টা করুন তো? এখন অব্দি যতটুকু গ্নু-লিনাক্স জানেন, তা দিয়ে চেষ্টা করে করে যদি এটা করে ফেলতে পারেন, তাহলে, এই মুহূর্তে একটা এপিঠ-ওপিঠ করান, নিজের নাম বদলে ফেলুন আইনস্টাইনে। কেন? কারণ, একটু আগেই যেমন আমরা দেখলাম, আগে হাইফেনটা থাকলেই সেটাকে অপশান বলে মনে করে নিচ্ছে touch বা rm বা mv বা cp। নিজেই চেষ্টা করে দেখুন না।

অনাইস্টাইনদের জন্যে পরিজ্ঞাতব্য তথ্য এ টাই যে, এই ফাইলটাকে সাইজ করার কায়দা আমি এই লেখায় বলে দেবনা, ইচ্ছে করেই, যাতে গ্নু-লিনাক্স ডকুমেন্টেশন ঘাঁটার অভ্যেশটা হয় আপনার। যা গ্নু-লিনাক্স শেখার একটা প্রাথমিকতম শর্ত। যদি নিজে নেট থেকে টেক্সট নামিয়ে নেন বা বই কিনে নেন, তাহলে তো ভালোই। নইলে জিএলটির সিডিটার জন্যে মেল করবেন, একশো টাকা করে নিচ্ছি আমরা, ডাউনলোড আর সিডি পোড়ানোর খরচ বাবদ, তাতে গ্নু-লিনাক্স সংক্রান্ত ছহাজারের উপর ওয়েব পেজ এবং ছশোর মত পিডিএফ ফাইল রাখা আছে, যার গোটাটা শেখা হয়ে গেলে আপনি অনায়াসে আন্তর্জাতিক খেতাব পরমলিনাক্সচক্রের জন্যে দরখাস্ত করতে পারেন, ওটা পেয়ে গেলে বিনা পয়সায় ইন্টারনেট করা যায়, আর সারা বছরই ঠ্যাং ভাঙার ছুটি দেয়। যারা দূরে থাকেন, লিখবেন, জিএলটি-র সিডি পোস্টে পাঠিয়ে দেব।

এবার একটা জিনিষ মাথায় রাখুন, যারা উইনডোজে কাজ করে অভ্যস্ত, তারা জানেন, উইনডোজ এক্সপ্লোরারে ঘুরে বেড়াতে বেড়াতে দেখা যায়, এক একটা বিশেষ ধরনের ফাইলের পাশে এক একটা বিশেষ ধরনের ছবি বা আইকন। এদের যে কোনোটায় মাউস দিয়ে ক্লিক করলেই একটা বিশেষ প্রোগ্রাম তাদের খোলে। যেমন যেসব ফাইল ‘*.doc’, তাদের যেকোনোটায় ক্লিক করলেই সেটাকে খুলবে এমএস-ওয়ার্ড, ‘*.xls’ হলে তাদের খুলবে এমএস-একসেল, ‘*.txt’ হলে খুলবে নোটপ্যাড, বা খুব বড় সাইজ হলে, উইন্ডোজ ওয়ার্ডপ্যাড। এদের সিস্টেম চিনছে কিন্তু ফাইল এক্সটেনশন দিয়েই, মানে ডস ফাইলনাম কাঠামোর ওই ৮.৩-এর পরের তিনটে অক্ষর দিয়ে। সিস্টেমের কিরকম বিকট গ্যাঁড়ামো দেখুন, আপনি যখন ৪ অক্ষরের কোনো এক্সটেনশন ব্যবহার করছেন, ‘glinux.html’, তখনো সিস্টেম তাকে চিনছে ওই ৮.৩-এই, আপনি এমএসডস প্রম্পটে গিয়ে ডির (dir) মেরে দেখুন, সিস্টেম দেখাচ্ছে ‘glinux~1.htm’। এই নামটা দিয়ে টাইপ কমান্ড মেরে দেখুন, ‘`type glinux~1.htm`’, দিব্য দেখবেন সোনামুখ করে ওয়েবপেজটার এইচটিএমএল কোডিং দেখিয়ে দিল। আবার আপনি উইনডোজ এক্সপ্লোরারে যান, দেখবেন ওই ফাইলটার পাশে ওয়েবপেজের আইকন ফুটে উঠেছে, এবং ক্লিক করলে তাকে ইন্টারনেট এক্সপ্লোরার দিয়েই খুলছে। এই কাজের নাম আর দেখানোর নামের কথা তো আমরা আগেই বলেছি। এবার মজাটা হল, আপনি কোনো একটা নামের এক্সটেনশন মানে পরের তিন অক্ষরের অংশটা বদলাতে যান, দেখবেন কী চেঁচায়, ‘তুমি এসব করলে যদি আর কখনো ফাইলটা খুলতে না-পারো, আমি কিন্তু কিছু জানিনা’, আসলে জানেনা ও নিজেই। এক্সটেনশন বদলে ক্লিক করে দেখুন, ফাইলটা ও খুলতেই পারবে না। এবং প্রত্যেকটা ব্যবহারকারীকেই নিজের মত নির্বুদ্ধি বলে ধরে নেয় উইনডোজ। গ্নু-লিনাক্স-এর মত নিরাপত্তা ব্যবস্থা না-থাকায় আমি আগে যা করে রাখতাম সেটা হল কোনো ‘*.doc’ বা ‘*.txt’ ফাইলের এক্সটেনশনটা বদলে কোনো সিস্টেম ফাইল এক্সটেনশন দিয়ে রাখা যেমন ‘*.drv’ বা ‘*.sys’ ইত্যাদি। কেউ যদি ফাইলটা পায়, এবং ওরকমই পাঁঠা হয়, খুলতেই পারবে না। এবং সিস্টেমের গ্যাঁড়ামো এখানেই শেষ নয়, নিজের এই বোকামিটা আবার গোপন করারও চেষ্টা করে উইনডোজ। ডিফল্ট সেটিং-এ করা থাকে ফাইলনাম এক্সটেনশন না-দেখানোর ব্যবস্থা। উইনডোজ এক্সপ্লোরারের ‘টুলস’ মেনুর ‘ফোল্ডার অপশান’-এ গিয়ে তবে অফ করে দিতে হয় ‘হাইড ফাইলনেম

এক্সটেনশন'। মানে লোকে দেখবে না, অথচ ও দেখবে, আর এক্সপ্লোরারে পাশে প্রোগ্রামের আইকনটা দেখাবে, যেন ওর কত বুদ্ধি, ম্যাজিক করে জেনে ফেলছে। কোন ফাইলটা কোন প্রোগ্রামের।

এই ম্যাজিকটা সত্যিই 'ম্যাজিক' নামেই সত্যি গ্নু-লিনাক্স-এর বেলায়। এখানে তো কোনো এক্সটেনশন-এর চক্কর নেই, তাহলে কোনো ফাইলকে তার নিজের প্রোগ্রাম কী করে চিনে নেবে এক্স-উইনডোজের ব্রাউজার, বা, আমি নিজে যখন ব্যবহার করতে চাইব কোনো ফাইল, সেটা কী কাঠামোর ফাইল কী করে বুঝব? এতেই ব্যবহার হয় 'ম্যাজিক'। আমরা সে কথায় পরে আসছি।

২.৩।। সাধারণ ফাইলের গঠন

একটা গ্নু-লিনাক্স সিস্টেমে ফাইল হয় নানা রকমের। এর পরের সাবসেকশনে আমরা জানব সেসব। এখানে ফাইলের গঠন বলতে আমরা রেগুলার বা সাধারণ ফাইলের গঠনই বোঝাচ্ছি। কারণ অন্য ফাইলের (ডিভাইস এবং ডিরেক্টরি) গঠন কারনেলের দ্বারা নির্দিষ্ট। রেগুলার ফাইলের গঠনের কিছু তফাতের কথায় আমরা পরে আসছি, যখন আমরা রেগুলার ফাইলদের মধ্যে প্রকারভেদের কথায় আসব। কিন্তু সেই গঠন ফারাকের পরও একটা রেগুলার ফাইলের গঠনের একটা মৌলিক বৈশিষ্ট্য থাকে প্রতি অপারেটিং সিস্টেমেই। সেই গঠনের মধ্যেও কিছু তফাত থাকে। গ্নু-লিনাক্স তথা যে কোনো ইউনিক্স বা উইনডোজে রেগুলার ফাইলের গঠনে একটা একদেশতা আছে। কিন্তু চিরকাল সব অপারেটিং সিস্টেমের রেগুলার ফাইলের আকার বলতে যে এটাই ছিল তা নয়। এবং এখনো যে রেগুলার ফাইলের অন্য কোনো গঠন হয়না তা নয়। কিন্তু সেগুলো আমাদের আলোচনায় আসবে না। আমরা এখানে গ্নু-লিনাক্সের রেগুলার ফাইলে বাইট রাখার কায়দা নিয়ে দু-একটা কথা বলে নেব।

যুগ থেকে যুগে, অপারেটিং সিস্টেম থেকে অপারেটিং সিস্টেমে আম রেগুলার ফাইলের গঠনের কায়দা এক নয়, একাধিক। তার মধ্যে একটা কায়দা হল যেখানে একটা ফাইল মানেই পরপর কিছু বাইটের সমষ্টি। যে বাইটগুলো আলাদা করে খুব ব্যক্তিত্ববান চরিত্রবান ইত্যাদি হতেই পারে, কিন্তু তাদের ব্যক্তিত্ব বা চরিত্র বিষয়ে অপারেটিং সিস্টেম কোনো খোঁজই রাখে না, শত চাইলেও কোনো ক্যারেকটার সার্টিফিকেট দিতে পারবে না। অপারেটিং সিস্টেমের কিছু এসে যায়না। অপারেটিং সিস্টেম শুধু একটা খোঁজই রাখে, তাদের বাইটবানতা। মানে, তার ভাণ্ডারে কত বাইট আছে, কোন বাইট, কোন এলাকার কত নম্বর বাইট, ইত্যাদি। বাইটগুলো অর্থময় হয়ে ওঠে ব্যবহারকারীর স্তরে, বাইটগুলো থেকে অর্থ তৈরি হয় ইউজার লেভেল প্রোগ্রামগুলোর হাতে। গ্নু-লিনাক্স তথা যে কোনো ইউনিক্স, এবং উইনডোজও, ফাইলগুলোকে এভাবেই দেখে। অপারেটিং সিস্টেমের কাছে ফাইলগুলো বাইট-সমাহারের চেয়ে বেশি কিছু না-হওয়াটা কাজের পক্ষে সবচেয়ে সহজ এবং নমনীয় একটা অবস্থা তৈরি করে। ইউজার লেভেল প্রোগ্রামগুলো তাদের যেভাবে প্রয়োজন যা প্রয়োজন তাই রাখতে পারে ফাইলে, নিজেদের কাছে সবচেয়ে সুবিধেজনক রকমে যা খুশি নাম দিতে পারে ফাইলের। অপারেটিং সিস্টেম যেমন কোনো সাহায্য করেনা তেমনি কোনো বাধাও দেয়না। আর কোনো ঝুঁকিপ্রবণ ব্যবহারকারী যখন তার নিজের রকমে অপারেটিং সিস্টেম নিয়ে পরীক্ষা-নিরীক্ষা বা খেলাধুলো করতে চায়, সেখানে এটা একটা জরুরি ব্যাপার হয়ে উঠতে পারে।

এছাড়া অন্য যে আকারের ফাইল হত বা আজো কোনো বিশেষ বিশেষ কাজের জন্যে হয়, তাদের আমরা আর আলোচনায় আনছি না। কৌতূহল মেটানোর জন্যে বলে রাখি, এক ধরনের ফাইলে তথ্য থাকত বিশেষ একটা নথি বা রেকর্ডের ছকে। যেমন আশি-কলাম পাঞ্চড কার্ডের যুগে, অনেক মেইনফ্রেম অপারেটিং সিস্টেমেই ব্যবহার হত এই ফাইল। এই ফাইলে তথ্য লেখা মানে আর একটা রেকর্ড যোগ করা, যাতে আছে আশিটা চিহ্ন বা ক্যারেকটার, কিম্বা পুরোনো একটা রেকর্ড উড়িয়ে সেখানে এই নতুন রেকর্ডটা গুঁজে দেওয়া। বা ১৩২ কলাম লাইন প্রিন্টারের জন্যে রেকর্ডের সাইজ হত ১৩২ টা ক্যারেকটারের। আবার এখান থেকে তথ্য পড়া মানে একটা একটা করে রেকর্ড পড়া। এখন আর এই ফাইলের ব্যবহার হয়না। কিন্তু অন্য আর এক রকমের ফাইল আছে যেখানে তথ্যটা থাকে গ্নু-লিনাক্স ফাইলের মত অসংগঠিত বিট সমাহারের আকারে নয়, সেখানে তথ্যের সবসময়ই একটা কাঠামো থাকে, একটা ট্রি বা ডালপালাসহ গাছের মত করে সাজানো থাকে তথ্যটা। এই ধরনের ফাইল এখনো ব্যবহার হয় কোনো কোনো বাণিজ্যিক ডেটাবেস নিয়ে কাজ করা মেশিনে। এই ধরনের ফাইলে তথ্য থাকে নির্দিষ্ট কিছু কি বা বিশিষ্ট বিন্দুকে কেন্দ্র করে, এবং যখন সেই বিশিষ্ট বিন্দুর বিষয়ে কোনো তথ্য খোঁজা হয়, তখন জানার দরকার পড়েনা ফাইলের কোন জায়গায় তথ্যটা আছে, ওই বিশিষ্ট বিন্দুর সাপেক্ষে সেটা অপারেটিং সিস্টেমই বার করে দেয়। অর্থাৎ এখানে ফাইলে তথ্য রাখা এবং পড়ার ব্যাপারে অপারেটিং সিস্টেম সরাসরি অংশগ্রহণ করে। কিন্তু এই দু ধরনের কোনো ফাইলই আমাদের প্রয়োজন পড়বে না, আমাদের কাছে ফাইল মানেই কিছু বাইটের সমষ্টি, যাদের বাইট কাঠামো ছাড়া আর কিছু অপারেটিং সিস্টেম জানেনা। সেসব জানে সেই প্রোগ্রাম যে ফাইল নিয়ে কাজ করে।

কিন্তু তাই যদি হয়, তাহলে, অপারেটিং সিস্টেম যদি নাই জানে কোন ফাইলে কী আছে, তাহলে এক নম্বর দিনে আমরা যে বিভিন্ন ধরনের ফাইলের পার্থক্যের কথা বললাম, কোনো ফাইল টেক্সটের, কোনো ফাইল কোডের, কোনো ফাইল প্রোগ্রামের, যাদের নাম লিখে এন্টার মারলেই চলতে শুরু করে, সেই তফাতটা ঘটছে কী করে?

২.৪।। রেগুলার ফাইল
একটা গ্নু-লিনাক্স সিস্টেমে তিন ধরনের ফাইল থাকে, রেগুলার বা সাধারণ ফাইল, ডিরেক্টরি ফাইল, এবং ডিভাইস ফাইল। এইমাত্র ফাইলের তথ্যের কাঠামো নিয়ে আমরা যে আলোচনাটা করলাম সেটা এই রেগুলার বা অর্ডিনারি বা সাধারণ ফাইল। কারণ, ডিরেক্টরি ফাইল আর ডিভাইস ফাইলের চরিত্র একদম ভিন্ন ধাঁচের। তার কথায় আমরা পরে আসছি। রেগুলার বা সাধারণ ফাইলে থাকে বিভিন্ন ধরনের ব্যবহারকারীর স্তরের তথ্য। এমনকি যখন সিস্টেমের কোনো প্রসেস এই রকম কোনো ফাইল লেখে, সেই মুহূর্তে সেই প্রসেসও একজন ব্যবহারকারী। যেমন পরে আমরা যখন ইউজার আইডি বা ব্যবহারকারীর পরিচিতি কাকে বলে জানব, দেখব, একটা গ্নু-লিনাক্স অপারেটিং সিস্টেমে ব্যবহারকারীদের তালিকায় নাম থাকে man, games, mail, news, postfix ইত্যাদি একাধিক সিস্টেম প্রসেসের। পাঁচ নম্বর দিনের আলোচনা থেকে /etc/passwd ফাইলটা মিলিয়ে দেখুন। যে কোনো ব্যবহারকারী না-হলেও, সর্বোচ্চ ব্যবহারকারী মানে রুট যে কোনো সময়েই এর যে কোনো ফাইলে কী আছে দেখতে পারে। অথচ কোনো ডিরেক্টরি ফাইল এমনকি সে-ও পড়তে পারেনা। ডিরেক্টরির মধ্যে কী কী ফাইল বা সাবডিরেক্টরি আছে সেই তালিকাটা দেখতে পারে, কিন্তু খোদ ওই ডিরেক্টরি ফাইলটা পড়তে পায়না। যদিও সে যে কোনো সময় যে কোনো ডিরেক্টরি ওড়াতে পারে বা নাম বদলাতে পারে। ডিরেক্টরি ফাইলরা হল সিস্টেম ফাইল, যারা সিস্টেমের কাঠামো ধরে রাখে।
রেগুলার বা সাধারণ ফাইলগুলোয় সচরাচর ভরা থাকে অ্যাসকি টেক্সট বা বাইনারি তথ্য। সিস্টেমের কিছু এসে যায়না, ফাইলে বাইনারি আছে না অ্যাস্কি আছে না গুষ্টির পিন্ডি আছে, এসে যায় ব্যবহারকারীর। সিস্টেম এই বাইনারি বা অ্যাসকির ভিতর কোনো তফাত করেনা, সবই তার কাছে রেগুলার ফাইল। শূন্য নম্বর দিনের আলোচনাগুলো মনে করুন, এই তথ্যের কাঠামো নিয়ে আমরা যেখানে কথা বলেছি। আর এক নম্বর দিনে আমরা বলে এসেছি বাইনারি, অ্যাস্কি ইত্যাদি বিভিন্ন ফাইলের কাজ কী? মনে করতে পারছেন? আগেই বলেছি, অ্যাসকি ফাইলে থাকে লাইন লাইন টেক্সট। এর সুবিধেটা এই যে খুব সহজে এদের স্ক্রিনে দেখা যায়, প্রিন্ট নেওয়া যায়, কোনো এডিটর দিয়ে এডিট করা যায়।
আর একবার মনে করিয়ে দিই — ওয়ার্ড প্রসেসর নয়, এডিটর। ওয়ার্ড প্রসেসর মানে ওয়ার্ডকে প্রসেস করে, প্রসেসিত ওয়ার্ডের নানা চিহ্ন গুঁজে দেয় নির্জলা, র, টেক্সট-এর মধ্যে, সিস্টেম তখন আর তাকে অ্যাসকি টেক্সট হিশেবে কাজে লাগাতে পারেনা। ধরুন, লিলো-কনফ ফাইলটা, এর প্রথম লাইনটাকে আপনি বোল্ড করেছেন আপনার ওয়ার্ড প্রসেসরে, প্যারাগ্রাফগুলোকে বাঁদিক থেকে আধ ইঞ্চি সরিয়ে এনে ইনডেন্ট করেছেন, এবার সেই ফাইলটায় আপনার ওয়ার্ড প্রসেসর এই সব করার আদেশগুলো নানা বিচিত্র রকমে এবং চিহ্নে গুঁজে দিল। এবার সিস্টেম যখন, বুট করার সময়ে, লিলো-কনফ পড়তে গেল, কোথায় কোন কারনেল আছে জানার জন্যে, সে ফাইলের মধ্যে গিয়ে পেল সেইসব দেবশিশুদের পদচিহ্ন, কিন্তু সিস্টেম তার বাপের জন্মে, ইউনিক্স মাল্টিক্স এফএমএস, কোনোদিন এসবের মানে জানেনা, পড়তে পারল না, এবং ভারি হিংসুটে সে, এটা হল কর্নিশ অগরের কাছে যাওয়ার আগের ফেজ, সিস্টেমের দৈত্য কিছুতেই বুট করতে রাজি হলনা, স্ক্রিনে ফুটে উঠল 'বুট ফেইলিওর . . . কারনেল প্যানিক'। প্যানিক শুধু কারনেলের, আপনার প্রাণও হঁক হঁক করছে। ভাবছেন, কী দরকার ছিল এসব অক্ষরের শব্দের প্যারার সৌন্দর্যবোধ ওই কেলে চাঁড়াল দৈত্যকে শেখাতে যাওয়ার, একি কালচারাল পাবলিক নাকি, সৌন্দর্য এবং বোধে গলে গলে পড়ছে। এর জন্যে দরকার কোনো ম্যাটার-অফ-ফ্যাক্ট এডিটর, যে নিখাদ টেক্সট আর লাইন-ভাঙার চিহ্ন ছাড়া আর কিছু গুঁজবে না ফাইলের মধ্যে, যেমন ইম্যাক্স, ভিম, জো, জেডিট, গেডিট, নেডিট, কেডিট ইত্যাদি। ওপেন অফিস রাইটার, অ্যাবি-ওয়ার্ড, কে-অফিস ইত্যাদি ওয়ার্ড প্রসেসর দিয়ে হবেনা।
এই অ্যাস্কি ফাইলগুলোকে শুধু যে এডিট করা যায় তাই নয়, যদি অনেক প্রোগ্রামই ইনপুট আউটপুট হিশেবে এই অ্যাস্কি ফাইল ব্যবহার করে, তখন এদের একটার আউটপুটকে অন্যটার ইনপুটে জুতে দেওয়া যায়, শেলের পাইপলাইন দিয়ে, পথনির্দেশ করে। একটু আগে দেখেননি, '`man -k CD | grep 'audio'> audio.cd.program.text`' কমান্ডে? ম্যান প্রথমে সমস্ত ম্যানুয়াল পাতা ঘেঁটে দেখল, কোথায় কোথায় সে পাচ্ছে '`CD`' শব্দটা। এই '`man -k CD`' কমান্ডের আউটপুট মানে সেই ৩৮ লাইন অ্যাস্কি আউটপুট চলে গেল গ্রেপ কমান্ডের কাছে ইনপুট হয়ে। '`grep 'audio'`'। গ্রেপ এবার তার সামনে পেল ওই আটত্রিশ লাইন ইনপুট, তাদের মধ্যে সে খুঁজে দেখল কোনো লাইনে '`audio`' শব্দটা আছে কিনা। এবার গ্রেপ কমান্ডের যে ছ লাইন অ্যাস্কি আউটপুট, যেটা আগে আমরা দেখছিলাম স্ক্রিনে, সেটা রিডাইরেক্ট বা চালান হয়ে গেল একটা অ্যাস্কি টেক্সট ফাইলে, যার নাম '`audio.cd.program.text`'।
সাধারণ বা রেগুলার ফাইলদের মধ্যে সবচেয়ে সংখ্যাগরিষ্ঠ এই অ্যাস্কি টেক্সট ফাইল। শূন্য নম্বর দিনে আমরা যে চিহ্ন বা ক্যারেকটারের কথা বলেছিলাম, যাদের স্ক্রিনে দেখা যায়, বা ছাপা যায়, তাদের দিয়েই ভরা থাকে এই ফাইলগুলো। ls, who, tar, less, wc, man, ইত্যাদি যে কমান্ডগুলো আমরা ব্যবহার করছিলাম তারা কিন্তু এই অ্যাস্কি ফাইল নয়, বা সি কোড লেখা কোনো ফাইল আপনি কম্পাইল করার পর যে একজিকিউটেবল বা চালানো-যায়-এমন বাইনারি ফাইল পাবেন, তারাও অ্যাস্কি টেক্সট ফাইল নয়। টেক্সট ফাইলের গোটা টেক্সটা ছোট ছোট দলে ভাঙা থাকে, এক একটা লাইনে এক এক দল চিহ্ন। প্রতিটি

লাইন শেষ হয় ‘লাইনফিড’ নামের চিহ্নে, অ্যাসকি সিস্টেমে যার ডেসিমাল বা দশমিক মান ১০, মনে করতে পারছেন? এই ‘লাইনফিড’-কে ‘নিউলাইন’ বলেও ডাকা হয়। এই লাইনফিড বা এলএফ চিহ্নটার বিশেষত্বটা দেখুন, এই চিহ্নটাকে কিন্তু আপনি দেখতে পাননা, না স্ক্রিনে না প্রিন্টে, দেখতে পান তার কাজের নমুনাটাকে। যেই একটা লাইন ভেঙে আর একটা নতুন লাইন শুরু হয়। টেক্সট লেখার সময় আপনি যেই এন্টার মারেন, অমনি সেখানে একটা লাইনফিড ঢুকে পড়ে। পরিচিত ওয়ার্ড-প্রসেসরগুলোয়, যখন আপনি অপশানে দেন ‘শো নন-প্রিন্টিং ক্যারেকটারস’, অদৃশ্য চিহ্নগুলো দেখাও, তখন অন্য অনেক বাড়তি চিহ্নের সঙ্গে এই লাইনফিড-কেও দেখায়, সচরাচর এর চিহ্নটা থাকে ‘¶’। কিন্তু একটা জিনিষ খেয়াল রাখবেন, ওয়ার্ড প্রসেসর এটাকে দেখায় শুধু আপনি যেখানে লাইন ভেঙেছেন, মানে এন্টার মেরেছেন, স্ক্রিনের বা প্রিন্ট-এর পাতার মাপের সঙ্গে মিলিয়ে মিলিয়ে লাইনকে সে ভেঙে দেয় নিজেই, যার টেকনিকাল নাম ওয়ার্ড-র‍্যাপ, যা একটা বিশুদ্ধ টেক্সট এডিটর করেনা, যদি আপনি আলাদা করে না-বলে দেন, এমনিতে সেখানে লাইন ভাঙে শুধু যখন আপনি ভেঙেছেন, বা লাইনফিড দিয়েছেন এন্টার মেরে।

আর অন্য ফাইল হল বাইনারি ফাইল, মানে, এক কথায় না-অ্যাস্কি রেগুলার ফাইল। স্ক্রিনে ক্যাট করলে বা প্রিন্ট নিলে ঠিক সেই জাতের জঞ্জাল তৈরি হয়, যার সাথে আমরা পরিচিত হয়েছিলাম, শূন্য নম্বর দিনে, ভিশুয়াল চ্যাংড়ামোয়। আমাদের কাছে মনে হয় বিশুদ্ধ এবং পরিণামহীন পরিচ্ছন্ন জঞ্জাল, কিন্তু ‘ls’ খুলে দেখেছি, প্রত্যেকবার আমরা যখন এলএস মারি, শেল তাকে খুঁজে বার করে ‘/bin/ls’ থেকে, সিস্টেম তাকে পড়ে ফেলে এবং এক্সিকিউট করে, কী বস্তু সে প্রতিবার পড়ে, এই কৌতূহল থেকে, সেই ফাইলে একদম প্রথম দিকেই অন্তত দুটো চেনা শব্দ খুঁজে পেয়েছিলাম, ‘gnu’ আর ‘/lib/ld-linux.so.2’। দ্বিতীয়টা ঠিক কী ধরনের চীজ কোনো আন্দাজ করতে পারছেন? কোনো আলগা আন্দাজ? আমাদের কাছে যেটা এলোপাথাড়ি জঞ্জাল বলে মনে হচ্ছে, অবভিয়াসলি তার মধ্যেও একটা কাঠামো আছে, মেশিন ভাষার কাঠামো। যা থেকে প্রোগ্রামটা কাজ করে। যদিও গোটাটাই হল পরপর কিছু বাইটের সমাহার, কিন্তু এর মধ্যেই একটা কিছু থাকে যা থেকে কোনো ফাইলকে সিস্টেম একজিকিউট করে, চালায়, অন্য প্রোগ্রামকে চালায় না, তার মানে সিস্টেম একে একজিকিউটেবল বলে চিনতে পারে।

একটা একজিকিউটেবল বাইনারি ফাইলের মূলত পাঁচটা খণ্ড থাকে — হেডার, টেক্সট, ডেটা, রিলোকেশন বিট, এবং সিম্বল টেবল। আর এই হেডার শুরু হয় সেই ম্যাজিক নাম্বার দিয়ে, যা থেকে সিস্টেম এই ফাইলটাকে একটা একজিকিউটেবল ফাইল বলে চিনতে পারে। এই ম্যাজিক নাম্বার দিয়ে বিভিন্ন ধরনের ফাইলকে চেনে সিস্টেম। গ্নু-লিনাক্সে একটা কমান্ড আছে, ফাইল (file), যা ফাইলের মধ্যেকার হেডারের ওই ম্যাজিক নাম্বার আর অন্য প্রয়োজনীয় তথ্য পড়ে আপনাকে জানিয়ে দেয় ফাইলটার সম্পর্কে আপনি যা জানতে চান। সিস্টেমও এই ভাবেই জানে, এর জন্যে কোনো ফাইল এক্সটেনশন জাতীয় কায়দা বা তাকে জানানোর জন্যে আরো আরো কায়দা তার দরকার পড়েনা। ‘file /bin/ls’ আর ‘file essay.1.text’ করে দেখুন তো কী দেখাচ্ছে। বা, যে কোনো ফাইলকে। ফাইলের কোনো ব্যবহারগত বৈশিষ্ট্য বা অ্যাট্রিবিউট, যেমন কার মালিকানা এই ফাইলের উপর, কে কে একে পড়তে বদলাতে বা চালাতে পারবে, এই জাতীয় কোনো তথ্যই গ্নু-লিনাক্স সিস্টেমে কোনো ফাইলের মধ্যে থাকেনা। এমনকি, হায়, ফাইল নিজে নিজে নিজের নামটা অব্দি জানেনা, সেটাও ফাইলের মধ্যে থাকেনা।

২.৫।। ডিরেক্টরি ফাইল

ডিরেক্টরি ফাইলে কোনো বাইরের তথ্য থাকেনা, শুধু তার মধ্যে যে ফাইল বা সাবডিরেক্টরিগুলো আছে, তাদের কিছু খুঁটিনাটি বিবরণ থাকে। গ্নু-লিনাক্স ফাইলসিস্টেম সাজানো থাকে এরকম ডিরেক্টরির মধ্যে ডিরেক্টরির মধ্যে ডিরেক্টরি দিয়ে। নিজের মালিকানার মধ্যে যেসব ডিরেক্টরি আছে, ঠিক ফাইলের মতো, যখন খুশি ওড়াতে পারে কোনো ব্যবহারকারী, আবার নিজের মালিকানার এলাকায় যেখানে খুশি যেমন খুশি ডিরেক্টরি বানাতে পারে। ডিরেক্টরি ওড়ানোর কমান্ড হল ‘rmdir’, আর বানানোর কমান্ড ‘mkdir’। কিন্তু, একটা ডিরেক্টরিকে ‘rmdir’ দিয়ে ওড়ানো যায় শুধু তখনই যখন তার পেটে আর কোনো ডিরেক্টরি বা সাবডিরেক্টরি বা ফাইল নেই। সেই অবস্থায় কমান্ড হল ‘`rm -fr`’। তবে এই কমান্ডটা একটা ভয়ঙ্কর জিনিষ। আপনি রুট হয়ে মানে সুপারইউজার হয়ে, যখন আপনার সব কাজের অনুমতি আছে, কমান্ড দেন ‘`rm -fr /`’, সিস্টেম ফাইল ওড়াতে শুরু করে দেবে। একদম গোড়া থেকে, প্রতিটি ডিরেক্টরির প্রতিটি সাবডিরেক্টরির প্রতিটি ফাইলকে উড়িয়ে যখন থামবে তখন আপনার সিস্টেম হল নিছক একটা শূন্য, হার্ড ডিস্কের আর সার্কিটের ভৌত অবয়ব। নতুন করে বুট করলে সিস্টেম বুটই করবেনা, দেখাবে ‘সিস্টেম রিকোয়ার্ড’। আমি সত্যিই করে দেখেছিলাম, সমস্ত প্রয়োজনীয় ফাইলের ব্যাকআপ সিডিতে নিয়ে, আর কোনোদিন কোনো কিছু ফেরত পাওয়া যায়না, এই শূন্য করে দেওয়াটা হয়, টেকনিকাল ভাষায় মিলিটারি ওয়াইপিং। যদি প্রতিটি সফটওয়ার প্রতিটি ডেটাফাইল নতুন করে পয়দা করার উপায় না-থাকে, সাবধান — শূন্য করে ভরে দেওয়া যাহার খেলা তারই সাজে।

ডিরেক্টরি বানাতে হয়, আমরা দেখেছি, যখন কিছু নির্দিষ্ট ধরনের ফাইলকে একই সঙ্গে রাখার দরকার পড়ে। ফাইলগুলোকে সংগঠিত করার দরকার পড়ে। একটা ডিরেক্টরিতে একই নামে একটার বেশি ফাইল থাকতে পারেনা, কিন্তু, আপনি যখন প্রবন্ধের সংকলনের পর সংকলন লিখে চলেছেন, নতুন নতুন ডিরেক্টরি, book.one, book.two, book.three ইত্যাদি, তার

প্রত্যেকটাতেই একটা করে টেবল-অফ-কন্টেন্ট মানে সূচীপত্র মানে toc.text থাকতেই পারে। প্রতিটি ডিরেক্টরিতে, ডিরেক্টরির মধ্যের প্রতিটি ফাইলের জন্যে রাখা থাকে দুটি করে তথ্য রাখার জায়গা বা ফিল্ড। একটা ফিল্ড হল ফাইলটার নাম, অন্যটা ফাইলটার আইনোড নম্বর। একটা ডিরেক্টরিতে যদি দশটা ফাইল থাকে, তার জন্যে থাকবে দশটা এন্ট্রি, প্রতিটিতে ওইরকম দুটো করে ফিল্ড। এই ডিরেক্টরি ফাইলেই লেখা থাকে তার ভিতরকার ফাইলগুলোর নাম, যে নাম আপনি দিয়েছেন, বা কোনো প্রোগ্রাম দিয়েছে। কিন্তু আগেই বলেছি, সরাসরি ডিরেক্টরি ফাইলে লেখার অধিকার কারনেল ছাড়া আর কারোর নেই, এমনকি রুটেরও না। কেউ কোনো ফাইল বা ডিরেক্টরি বদলানো মাত্র সেই ডিরেক্টরির ফাইলে সেই বদলটা লিখে দেয় কারনেল। এইগুলো নিয়ে আরো ভালো করে আলোচনায় আসছি আমরা ফাইল বৈশিষ্ট্য বা অ্যাট্রিবিউট-এর আলোচনায়।

২.৬।। ডিভাইস ফাইল

আমরা দু-নম্বর দিনের ৫ নম্বর সেকশনে ব্লক স্পেশাল এবং ক্যারেকটার স্পেশাল ডিভাইসের কথা বলেছিলাম, সেই দুরকম ডিভাইসের জন্যে গ্নু-লিনাক্সে থাকে দু-রকমের ফাইল — ব্লক স্পেশাল এবং ক্যারেকটার স্পেশাল। আগেই তো বলেছি, গ্নু-লিনাক্সে সবকিছুই এক একটা ফাইল। হার্ড ডিস্ক, প্রিন্টার, সিডি ড্রাইভ, কনসোল, মোডেম, মেমরি — সবকিছুই এক একটা ফাইল। ক্যারেকটার স্পেশাল ফাইল বরাদ্দ থাকে ক্যারেকটার ডিভাইসের জন্যে — অক্ষর বা চিহ্ন বা ক্যারেকটারের ইনপুট এবং আউটপুট করাই যার কাজ, আমি যখন আমার কম্পিউটার থেকে মেল করছি কোনো ইমেল আইডিতে, আমার সিস্টেম কিছু বিশেষ চিহ্ন লিখছে একটা বিশেষ ফাইলে, আউটপুট করছে, যে ফাইলের নাম /dev/modem, মানে আমার চিঠিটা মোডেম বেয়ে নেটে পাঠিয়ে দিচ্ছে। আবার যখন আমি নেট থেকে একটা ওয়েবপেজ পড়তে চাইছি, তখন আবার মেশিন নেট থেকে মোডেমে আসা তথ্য পড়ে ফেলছে ওই একই /dev/modem ফাইল থেকেই, এবার কিছু চিহ্ন বা অক্ষর বা ক্যারেকটারের ইনপুট। ব্লক স্পেশাল ফাইল থাকে হার্ড ডিস্ক ইত্যাদি ব্লক ডিভাইসের জন্যে। নির্দিষ্ট মাপের ব্লক থাকে একক হিশেবে, সেই ব্লকে ব্লকে এখানে তথ্য ইনপুট আউটপুট করা হয়। যখন আমি আমি আমার দ্বিতীয় হার্ড ডিস্কের তিন নম্বর পার্টিশনে কোনো তথ্য রাখছি, বা সেখান থেকে পড়ছি, সেই কাজটা ঘটছে /dev/hdb3 নামের একটা ব্লক ডিভাইস ফাইল দিয়ে। এই ব্লক ডিভাইস এবং ক্যারেকটার ডিভাইস ফাইল এই দুইরকম ফাইলকে মিলিয়ে আমরা এক কথায় বলি ডিভাইস ফাইল। বলুন তো, র‍্যাম বা র‍্যান্ডম অ্যাকসেস মেমরিটা ঠিক কী ধরনের ডিভাইস ফাইল? ব্লক স্পেশাল না ক্যারেকটার স্পেশাল?

এই যে, প্রতিটি ভৌত উপাদানই এক একটা ডিভাইস ফাইল, হার্ড ডিস্ক ড্রাইভ, সিডি ড্রাইভ, ফ্লপি ড্রাইভ, কনসোল বা টার্মিনাল, প্রিন্টার, মোডেম — সমস্ত কিছু — প্রথম প্রথম এটা একটু ঘোলাটে লাগতে পারে, কিন্তু পরে দেখবেন, বেড়ে সুবিধে আছে এর, যেসব কমান্ড দিয়ে কব্জা করা যায় একটা রেগুলার ফাইলকে, সেগুলো ডিভাইস ফাইলের উপরও ব্যবহার করা যায়। যেমন, ক্যাট, কনক্যাটেনেট করে, মানে, স্ক্রিনের ডিসপ্লেতে পেতে দেয় একটা ফাইলকে, বা স্ক্রিন ছাড়া যেখানে চাইবেন, অন্য কোনো ফাইলেও ক্যাট করতে চাইতে পারেন একটা ফাইলকে। আপনি '`cat essay`' কমান্ড দিলে essay ফাইলটাকে স্ক্রিনে পড়ার জন্যে পেতে দেয় ক্যাট। মজার কথা হল, এই একই ভাবে আপনি মাউস ডিভাইস বা /dev/mouse ফাইলটাকেও ক্যাট করতে পারেন। '`cat /dev/mouse`' কমান্ড দিয়ে এন্টার মারুন, এবং দেখুন কমান্ড প্রম্পট ফেরত আসেনি, মানে আপনার আদেশ পালনের কাজটা শেষ হয়নি, চলছে। এবার মাউস নেড়ে দেখুন, প্রফেসর হিজিবিজবিজের কিছু যুগান্তকারী গবেষণা আপনার সামনে ভেসে উঠবে। ক্যাট করতে পারেন সরাসরি আপনার ভৌত হার্ড ডিস্কের একটা ভৌত পার্টিশনকে, করে দেখুন '`cat /dev/hda1`'। খুব মন খারাপ লাগবে, কত বুদ্ধিদীপ্ত প্রবন্ধে ভরে রেখেছেন হার্ড ডিস্ক, তারা আসলে এই? আরো একটা মজার খেলা হয়, কারনেলের কণ্ঠস্বর শোনা। /dev/dsp হল সাউন্ডকার্ড, কথাটা এসেছে ডিজিটাল সিগনাল প্রসেসর থেকে। আর পাঁচ নম্বর দিনে বলেছি, /boot/vmlinuz হল কারনেল। আপনার সিস্টেমের কার্ড কনফিগারেশন যদি ঠিক হয়ে থাকে তাহলে আপনি '`cat /dev/hda1 > /dev/dsp`' কমান্ড দিয়ে কারনেলকে সশব্দ শুনতে পাবেন।

ডিভাইস ফাইলটা এমনিতে কিন্তু ফাঁকা। আপনি তো ফাইলের সাইজ দেখতে জানেন, '`ls -sh /dev/hda1`' বা '`ls -sh /dev/mouse`' মেরে দেখে নিন, দুটোই দেবে শূন্য। অথচ ক্যাট দিলে দেখাচ্ছে। ঠিক এটাই মজা ডিভাইস ফাইলের। ডিভাইস ফাইলরা একদম শেখভের ডার্লিং-এর মত, যখন ভালোবাসে কোনো নাটকের ম্যানেজারকে, তখন তার ভারি দুঃখ, মানুষ কেন ভালো নাটক দেখতে শিখছে না, আবার কোনো কাঠব্যবসায়ীকে বিয়ে করার পর, গির্জে যাওয়ার পথে কিছুতেই রাগ সামলাতে পারেনা, এত যে দেরিতে বর্ষা নামল, ওক কাঠের দর এবার কত পার্সেন্ট বাড়বে বুঝতে পারছো, এদিকে মানুষ শুধু বাজে খরচ করছে থিয়েটার দেখে। ডিভাইস ফাইলের নিজের শরীরে কোনো তথ্যই থাকেনা, সব তথ্যই আসে তার দোসর মানে ভৌত ডিভাইসের কাছ থেকে। তেমনি, নিজে কিছু ধরেও রাখেনা, সবই পাঠিয়ে দেয় ওই ভৌত উপাদানকে। আপনি কোনো ফাইল পাঠালেন '`cp essay /dev/lpt1`' করে, মানে, আপনার 'essay' নামের ফাইলটা '/dev' ডিরেক্টরির 'lpt1' নামের ডিভাইস ফাইলে কপি হতে, ঠিক যেভাবে আপনি একটা রেগুলার ফাইলকে অন্য একটা নামে অন্য কোনো

জায়গায় কপি করেন, ওমা, দেখলেন পুরো শ্রমজীবী কোলাহল নিয়ে আপনার পাঠানো ফাইলটা প্রিন্ট হতে শুরু করেছে 'lpt1' নামের প্রিন্টার-পোর্টে লাগানো ডট-ম্যাট্রিক্স প্রিন্টার থেকে। শ্রমজীবী ঘটাং ঘটাং এর জায়গায় ওটা এলিট গুজগুজ ফুসফুস-ও হতে পারে, যদি ডেস্কজেট বা লেজারজেট হয়। এবং আপনার কমান্ড প্রম্পট সঙ্গে সঙ্গেই ফেরত এসেছে, মানে, সিস্টেমের কপি করা হয়ে গেছে, এখন যা ঘটার সব প্রিন্টারে ঘটছে। বা, ফ্লপির ডিভাইস ফাইলে আপনি কপি করছেন মানে ফ্লপিতে রাখছেন ফাইলটা। আপনি ওই কপি (cp) বা মুভ (mv) কমান্ড দিয়েই খালাস, পরের টুকু নিয়ে মাথা ঘামাচ্ছে কারনেল।

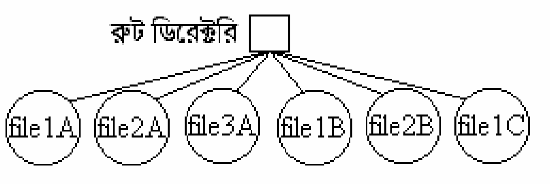
২.৭।। এক খাবলা পুনশ্চ

আপাতত আমাদের ফাইল কাকে বলে জানলাম। এবার আমরা যাব ডিরেক্টরি সিস্টেম নিয়ে আলোচনায়, যেখানে আমরা পথনির্দেশের খুঁটিনাটি জানব, কী ভাবে একটা ফাইলের ঠিকানা দেওয়া হয় একটা গ্নু-লিনাক্স সিস্টেমে। তার আগে, দুটো কথা বলে নেওয়ার আছে। ওই '`-essay`' নামের ফাইলটাকে কিছু করার উপায় না-দেওয়ায় আমার দুজন প্রাথমিক পাঠক খুব চেঁচানোয় উত্তরটা দিয়েই রাখছি। এই ধরনের যে কোনো ফাইলকে কব্জা করার উপায় হল মধ্যে একটা '`--`' দিয়ে নেওয়া। মধ্যে দেওয়া ওই '`--`' অংশটা শেলকে বলবে পরের অংশটাকে আক্ষরিকভাবেই একটা গোটা নাম হিশেবে পড়তে। কেন, সেটা আমরা জানতে পারব ব্যাশ শেল নিয়ে বিশদ আলোচনায়, যদি সেটা আদৌ আমাদের পাঠমালায় আসে অতটা দূর অব্দি। এবার যে কোনো কমান্ডেই মধ্যে এই '`--`' ডান্ডাটা দেওয়া থাকলে আর কোনো ঝামেলা হবেনা। যেকোনো কমান্ডই কাজ করবে। তবে, সবচেয়ে সহজ হল '`mv -- -essay essay`' করে ফাইলটার নামের হাইফেনটা উড়িয়ে দেওয়া। এখানে বলে রাখি, শুধু হাইফেন না, এই রকম ঝামেলা হতে পারে '$', '`', '?' বা '*' চিহ্ন নিয়েও। এই চিহ্নগুলোর শেলে আলাদা অর্থ আছে। তাই, এই চিহ্নগুলো ফাইলনামে ব্যবহার করলে শেল কুপথে চালিত হওয়ার সম্ভাবনা থাকে। নইলে বারণ নেই এর কোনোটা ব্যবহারেই। যে কোনো ল্যাটিন মানে ইংরিজি অক্ষর, তার বড় বা ছোট হাতে, 'A-Z' বা 'a-z', যে কোনো অঙ্ক, '0-9', পিরিয়ড মানে ডট বা বিন্দু বা '.', আন্ডারস্কোর মানে '_' — এর কোনোটাতেই কোনো সমস্যা নেই।

৩.১।। এক স্তরের সরলতম ডাইরেক্টরি সিস্টেম

এক একটা সিস্টেমে মোট থাকে হাজার হাজার লাখ লাখ ফাইল। আমার দুটো হার্ড ডিস্কে, মোট সাতটা পার্টিশন, তার মধ্যে একটা সোয়াপ ফাইল, মোট আশি জিবিতে, ফাইলের সংখ্যা ৫৫৬৯১৩। একটা ব্যক্তিগত সিস্টেমেই এই সাড়ে পাঁচ লাখেরও বেশি ফাইল, এবার একটা অফিস বা ইনস্টিটিউট-এর মোট ফাইলের সংখ্যা ভাবুন, এই অগণিত ফাইল থাকার জন্যে, অবশ্যই একটা ডাইরেক্টরি কাঠামো দরকার পড়ে। চার নম্বর দিনের সেকশন ১৩-য় আমাদের আলোচনা থেকে, ১৯৬০-এর দশকের পৃথিবীর সবচেয়ে শক্তিশালী মেইনফ্রেম কম্পিউটার সিডিসি ৬৬০০-র কথা মনে করুন, যাকে 'সুপারকম্পিউটার' বলে ডাকা হয়েছিল, যেখানে যাবতীয় সমস্ত ব্যবহারকারীর জন্যে ডিরেক্টরি ছিল সাকুল্যে একটা — প্রতিটি ইউজারের প্রতিটি ফাইলই থাকত সেই একমেবাদ্বিতীয়ম ডিরেক্টরিতে। ভাবুন তো সেইভাবে যদি রাখতে হত একটা ব্যক্তিগত সিস্টেমেরই ছয় লাখ ফাইল, তাহলে লোকে কাজ করবে কী, ফাইলের আলাদা আলাদা নাম আবিষ্কার করতে গিয়েই তো কাঁধ ঝুলে যাবে। আসলে সিডিসি ৬৬০০-র পরের চল্লিশটা বছরে পৃথিবী, মানুষ, তার কাজ, কাজের হাতিয়ার — সবই বদলে গেছে বিরাট রকমে। তাই এখন ডাইরেক্টরি কাঠামোও দরকার পড়ে অনেক বিশদ জটিল এবং স্তরবিন্যস্ত।

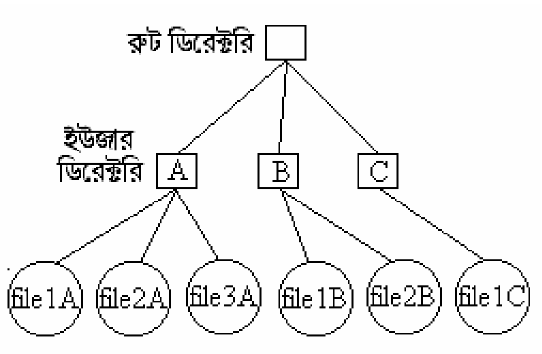
সিডিসি ৬৬০০ গোছের মেশিনে, এবং এর পরেও কিছুদিন অব্দি, ব্যবহৃত হত, এই সরলতম ডিরেক্টরি কাঠামো, যাতে একটাই ডিরেক্টরির উদরে সুখে বসবাস করত আপামর সমস্ত ব্যবহারকারীর যাবতীয় ফাইল। পরবর্তী ডিরেক্টরি কাঠামোর সঙ্গে তুলনা করে আলোচনা করতে গিয়ে এটাকেও রুট ডিরেক্টরি বলে ডাকা হয়, কিন্তু এক ভাবে দেখলে সেটা অর্থহীন। কারণ ডিরেক্টরিই তো সাকুল্যে এক পিস। এখানে এই ডিরেক্টরিটায় আছে ছটা ফাইল।



বাঁদিক থেকে প্রথম তিনটের মালিক ব্যবহারকারী A, ফাইলগুলো হল file1A, file2A, এবং file3A। পরের দুটো ফাইলের মালিক B, ফাইলদুটো file1B এবং file2B। ছ নম্বর ফাইল file1C, তার মালিক C। এখানে আমরা ফাইলের নাম না লিখে শুধু মালিকের নাম লিখলেই পারতাম, কারণ, ফাইলের নামটার এখানে কোনো গুরুত্বই নেই তেমন। একজন ব্যবহারকারীর কটা ফাইল আছে, এবং কত নম্বর ফাইলটা আমরা খুঁজছি, এটাই জরুরি। এবং এখানে সেই খুঁজে পাওয়াটা হবে অসম্ভব দ্রুত, কারণ খোঁজার জায়গাটা তো মাত্র এক।

এই ধরনের ডিরেক্টরি কাঠামোয় বিপজ্জনক জায়গাটা এই যে, কোনো একটা নামে আমি আমার একটা ফাইল লিখেছি, এবার আপনি যদি ওই একই নামে কোনো ফাইল সেভ করেন, সেটা আমার ফাইলটাকে ওভাররাইট করে, সেটার জায়গায় নিজেকে সিস্টেমে ঢুকিয়ে দেবে। এই ধরনের ডিরেক্টরি কাঠামো আমাদের পিসি সিস্টেমে আর ব্যবহার হয়না, হওয়া সম্ভবও না, কিন্তু

কোনো কোনো এমবেডেড সিস্টেমে, যেখানে প্রতিদিন গ্নু-লিনাক্সের ব্যবহার আগের চেয়ে বেড়ে যাচ্ছে, এটা ব্যবহার হওয়া সম্ভব। ধরুন একটা যন্ত্রের নিরাপত্তা ব্যবস্থা — কে কে সেটা ব্যবহার করতে পারবে, এখানে কয়েকজন ব্যবহারকারীর প্রত্যেকের একটা করে প্রোফাইল ভরে রাখতে হচ্ছে, পরিচয়ব্যবস্থা, হয়তো তাতে ছবি থাকছে, আঙুলের ছাপ থাকছে, সই থাকছে — ইত্যাদি। মূল ব্যাপারটা হল, প্রতিজন পিছু একটা করে ফাইল এবং সেই ফাইলের জায়গায় অন্য কারুর ফাইল সেভ করে দেওয়ার কোনো সম্ভাবনা নেই। যেমন আজ, ১২-ই নভেম্বর, খবরের কাগজে দেখছিলাম, বিভিন্ন জায়গায় এটিএম, বা অটোমেটিক টেলর মেশিন মূল কথাটা, কিন্তু কিরকম করে জানি এনিটাইম-মানি কথাটাই চালু হয়ে গেছে, সেই এটিএম ব্যবস্থা ভেঙে পড়ছে ভাইরাস আক্রমণে, এবং এই ব্যাপারে কথা বলতে গিয়ে একাধিক বিশেষজ্ঞ বলেছেন, এটা উইনডোজে কাজ হচ্ছে বলেই চিন্তা।



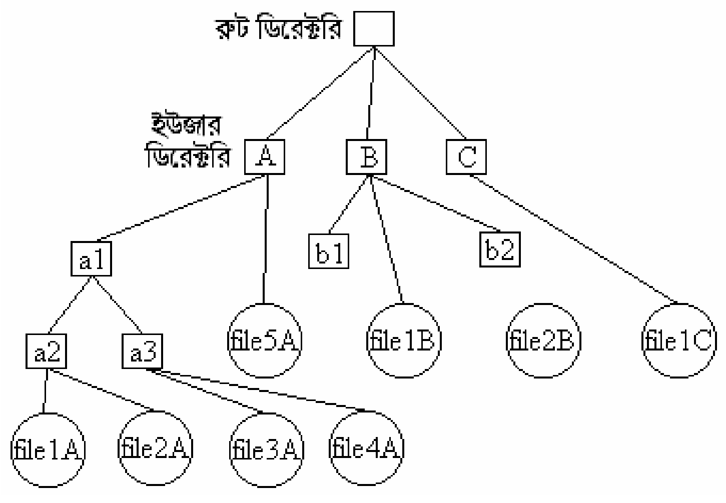
৩.২।। দুই স্তরের ডিরেক্টরি কাঠামো

এর পরের স্তরে, বিভিন্ন ব্যবহারকারীর মধ্যে গোলযোগ এড়াতে, এল দুই স্তরের ডিরেক্টরি কাঠামো। এই ছবিটাতেও দেখুন, আমরা চৌকো দিয়ে ডিরেক্টরি আর গোল দিয়ে ফাইল বুঝিয়েছি। এই দুই-স্তরের ডিরেক্টরি কাঠামোয় প্রতিটি ইউজার বা ব্যবহারকারীকে দেওয়া হল তার নিজের নিজের ডিরেক্টরি, যাকে আমরা A, B আর C দিয়ে দেখিয়েছি, তিন জন ব্যবহারকারীর নামে। এখন দেখুন প্রথম জনের তিনটে ফাইল, দ্বিতীয় জনের দুটো ফাইল আর তৃতীয় জনের একটা ফাইল আছে এদের নিজের নিজের ডিরেক্টরিতে। তাই ফাইলনাম এক না আলাদা তাতে আর কিছু এসে যায়না। একাধিক ইউজারের কম্পিউটারে, বা, খুব সরল নেটওয়ার্ক সিস্টেমেও এই কাঠামো ব্যবহার করা যায়।

এই ধরনের ডিরেক্টরি কাঠামোয় সিস্টেমের একটা ভূমিকা চলে এল এই অর্থে যে, যখনই কোনো একজন ব্যবহারকারী একটা ফাইল খোলার চেষ্টা করবে, সিস্টেমের জানা থাকবে, এই ব্যবহারকারী কে, এবং সেই অনুযায়ী তার মালিকানার ডিরেক্টরিতে গিয়ে ফাইলটা খুঁজবে। তার মানে, খেয়াল করুন, এইবার লগ-ইন ধারণাটা আসছে। এর সঙ্গে মেলান দিন চার-এ আমাদের মাল্টিপ্রোগ্রামিং থেকে মাল্টিইউজারে বিবর্তনের আলোচনাটা। লগ-ইন থাকছে মানেই এখানে প্রতিটি ব্যবহারকারীর একটা করে আইডেন্টিটি বা পরিচিতি এবং সেই পরিচিতি প্রমাণের দায় থাকছে, যা এক স্তরের ডিরেক্টরি কাঠামোয় ছিলনা।

এই স্তরের ডিরেক্টরি কাঠামোয় অন্য আর একটা ধরনের প্রয়োজন তৈরি হতে শুরু করল। ধরুন, প্রতিটি ইউজার শুধু তার নিজের ডিরেক্টরির ফাইলই খুলতে পারে, অন্য কোথাও কোনো ফাইলে তার অধিকার নেই। এই অবস্থায় যে সাধারণ কাজগুলো প্রতিটি ব্যবহারকারীকেই করতে হয়, তার প্রোগ্রামগুলোকেও প্রতিটি ব্যবহারকারীর ডিরেক্টরিতেই থাকতে হবে। আমরা যে প্রোগ্রামগুলোকে দেখেছি, /bin ডিরেক্টরিতে থাকতে, ls rm cp ইত্যাদি, সেগুলোর একটা করে কপি থাকতে হবে প্রতিটি ব্যবহারকারীর ডিরেক্টরিতেই। এটা একটা সিস্টেমের ইনএফিশিয়েন্সি, ক্যাবলামো। কাজের পরিমাণ, প্রোগ্রামের পরিমাণ যত বাড়ে এই ক্যাবলামোটা ততই আরো ব্যয়বহুল হয়ে পড়ে। অর্থাৎ, এমন একটা সিস্টেম ডিরেক্টরির দরকার পড়ছে যার মধ্যে থাকবে একজিকিউটেবল বাইনারি প্রোগ্রামগুলো — যা যে কোনো ইউজার ব্যবহার করতে পারবে।

৩.৩।। ক্রমানুসারী ডিরেক্টরি কাঠামো



এবার এল হায়েরার্কিকাল ডিরেক্টরি স্ট্রাকচার বা ক্রমানুসারী ডিরেক্টরি কাঠামো। দুই স্তরের কাঠামোয় নাম নিয়ে ঝামেলাটা চলে যায়, কিন্তু বহু ইউজারের বহু ফাইলের একটা ব্যবস্থায় খুব একটা সুবিধে হয়না। এমনকি একটা ব্যক্তিগত পিসি সিস্টেমেও খুব একটা কাজের নয়, আরো কাজের পরিমাণটা তো প্রতিদিনই বাড়ছিল। ইউজার তার কাজের যুক্তি মাফিক তার নিজের ফাইলগুলোকে আলাদা আলাদা ডিরেক্টরিতে রাখতে চাইবেই, যেমন

আপনার ওই প্রবন্ধ আর বইয়ের চক্করটা — এই রকম প্রয়োজন ক্রমে তো বাড়তেই থাকে, লেখার ফাইল, গানের ফাইল, কোডের ফাইল, কম্পাইল করার পর বাইনারি একজিকিউটেবল ফাইল, ইমেল ফাইল, ওয়েবপেজ, এইরকম। এই হায়েরার্কিকাল ডিরেক্টরি কাঠামোয় সেটা করার সুযোগ এল তার। এটায় দেখুন, সিস্টেমের বরাদ্দ করা তিনটে ইউজার ডিরেক্টরিকে আমরা দেখিয়েছি তিনজন ব্যবহারকারী — A, B আর C — তাদের নিজেদের নামেই। আর তাদের নিজেদের তৈরি করা ডিরেক্টরিগুলোকে দেখিয়েছি ছোট হাতে, a1 a2 a3 b1 b2 — এইরকম। এখানে দেখুন, A তার নিজের ইউজার ডিরেক্টরির মধ্যে a1 ডিরেক্টরি তৈরি করেছে। a1-এর মধ্যে আবার সাবডিরেক্টরি a2 আর a3। যার প্রত্যেকটায় দুটো করে মোট চারটে ফাইল। A-র পাঁচ নম্বর ফাইলটা আছে সরাসরি ইউজার ডিরেক্টরিতেই। ঠিক এই ভাবে যে কোনো ব্যবহারকারী তার নিজের ইউজার ডিরেক্টরিতে এবং তার মধ্যের যে কোনো সাবডিরেক্টরিতে যত খুশি ডিরেক্টরি এবং ফাইল বানাতে বা রাখতে পারে। একটা আধুনিক কম্পিউটার ব্যবস্থায় এই হায়েরার্কিকাল ডিরেক্টরি কাঠামোই ব্যবহৃত হয়।

৪।।পথনির্দেশ

এবার ডাইরেক্টরি কাঠামোয় সাজানো একটা ফাইল ব্যবস্থায় আমরা একটা জায়গা থেকে অন্য জায়গায় যাবার পথনির্দেশ করব কী করে? এর দুটো আলাদা উপায় আছে। একটা হল চূড়ান্ত পথনির্দেশ বা অ্যাবসলিউট পাথ। আর অন্যটা আপেক্ষিক পথনির্দেশ বা রিলেটিভ পাথ। ধরুন, এতক্ষণ ধরে, কোনো ডিরেক্টরি বলতে হলেই আমরা যে শুধু তার নাম না বলে সঙ্গে একটা স্ল্যাশ জুড়ে দিচ্ছিলাম, এই আগের আগের প্যারাগ্রাফেই দেখুন বাইনারি ফাইলগুলো কোথায় থাকে বলতে গিয়ে, 'bin' না-বলে, বললাম '/bin'। কেন? এটা বোঝাতে যে ওই 'bin' ডিরেক্টরিটা আছে '/'-এর মধ্যে, যা হল একটা গ্নু-লিনাক্স কাঠামোয় রুট ডিরেক্টরি, মানে আর সমস্ত ডিরেক্টরিরাই যার সাবডিরেক্টরি, বা সাবডিরেক্টরির সাবডিরেক্টরি। চূড়ান্ত পথ বা অ্যাবসলিউট পাথটা শুরুই হয় এই রুট ডিরেক্টরি স্ল্যাশ '/' থেকে। আবার এই স্ল্যাশ-ই ব্যবহার হয় একটা মা-ডিরেক্টরির সঙ্গে তার ছানা-ডিরেক্টরির সম্পর্ক বোঝাতে। যেমন ধরুন, আমরা বললাম '/usr/local/bin'। এর মানে, মূল মা ডিরেক্টরি বা রুট ডিরেক্টরি '/', তার ছানা হল 'usr', তার ছানা 'local', তার ছানা 'bin', সেই ডিরেক্টরির কথা বলছি আমরা — যার চূড়ান্ত ঠিকানা '/usr/local/bin'। এবার, যেই চূড়ান্ত ঠিকানাটা দিয়ে দিলাম, '/usr/local/bin' আর '/bin' আলাদা ডিরেক্টরি বলে চেনা গেল, দুজনের একই নাম 'bin' হওয়া সত্ত্বেও।

চূড়ান্ত বা অ্যাবসলিউট পথ তাই সবসময়ই শুরু হয় চূড়ান্ত বিন্দু বা মেরু থেকে, মানে রুট ডিরেক্টরি, গ্নু-লিনাক্সে যাকে আমরা লিখি স্ল্যাশ (/) দিয়ে, আর ডিরেক্টরি থেকে সাবডিরেক্টরি ভিন্নক বা সেপারেটরও ওই স্ল্যাশ। যাবতীয় ইউনিক্স মেক রিমেক মিক্স রিমিক্সেই এটা প্রথা। উইন্ডোজের সেপারেটরটা হল ব্যাকস্ল্যাশ (\), মাল্টিক্স-এ ছিল 'গ্রেটার দ্যান' বা 'বৃহত্তর' চিহ্ন (>)। এবার ধরুন, আমরা বলছি, '' ডিরেক্টরির মধ্যে আছে '' ডিরেক্টরি, তার মধ্যে আছে '' ডিরেক্টরি, তার মধ্যে আছে '' ডিরেক্টরি, যার কথা আলোচনা করছি আমরা। এই পথনির্দেশটা গ্নু-লিনাক্স তথা যেকোনো ইউনিক্স, উইনডোজ আর মাল্টিক্স, এই তিন রকম প্রথায় লেখা হত তিন ভাবে —

গ্নু-লিনাক্স — /usr/local/share/doc/

উইনডোজ — \usr\local\share\doc\

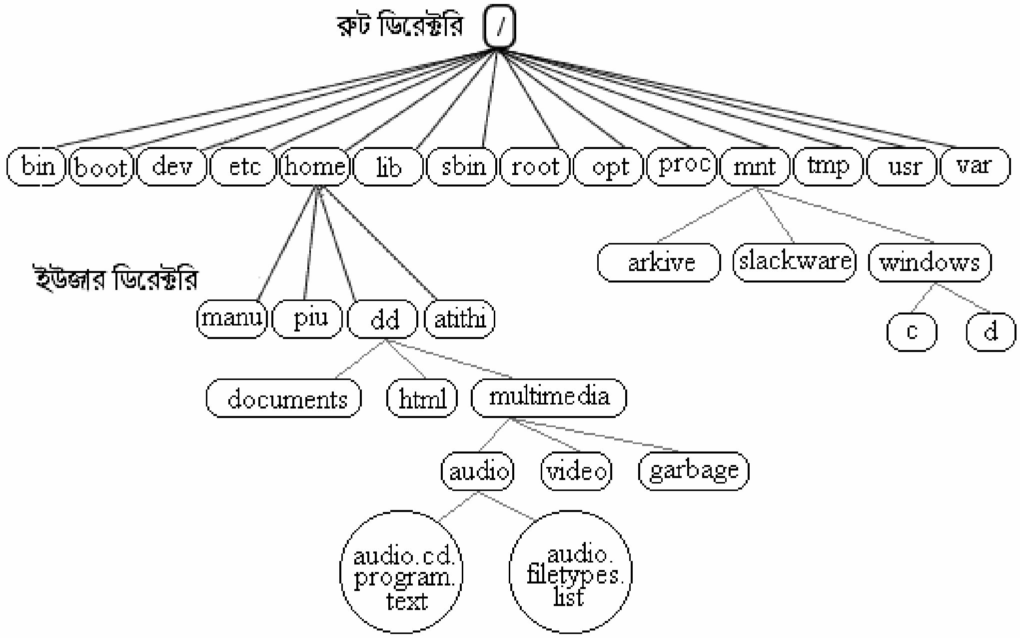
মাল্টিক্স — >usr>local>share>doc>

এখানে উইনডোজ পাথটা কিন্তু কোনো চূড়ান্ত পথ বা অ্যাবসলিউট পাথ নয়, কারণ, উইনডোজ '\' থেকে শুরু হয়না। উইনডোজ থেকে গ্নু-লিনাক্সে রুট ডিরেক্টরির ধারণাটা একদম আলাদা, সেটা আমরা পরে আরো ভালো করে বুঝতে পারব, মাউন্ট বোঝার সময়, এমনকি যদি 'C:' তার রুট ডিরেক্টরি বলে ধরে নিই, তাহলেও সেটা এখানে উল্লেখিত নেই। কিন্তু, গ্নু-লিনাক্স-এর বেলায় '/' দিয়ে শুরু মানেই সেটা অ্যাবসলিউট বা চূড়ান্ত পথ। সরাসরি রুট ডিরেক্টরি থেকে পুরো পথটা দেখাচ্ছে — যে অব্দি আমরা যেতে চাই।

অন্য রকমের পথনির্দেশটা হল আপেক্ষিক বা রিলেটিভ। আপেক্ষিক মানেই তার মধ্যে নিহিত আছে 'ওয়ার্কিং ডিরেক্টরি' বা 'কাজের ডিরেক্টরি' বলে একটা ধারণা, আপেক্ষিক বলতে সেই ওয়ার্কিং ডিরেক্টরির সাপেক্ষে আপেক্ষিক। এই শব্দবন্ধ 'ওয়ার্কিং ডিরেক্টরি' একবারই এসেছে আমাদের এই লেখায়, কোথায় মনে করতে পারছেন? একটা কমান্ডে, যে কমান্ডটা আপনি ব্যবহার করেছেন? কমান্ডটা ছিল, 'প্রিন্ট ওয়ার্কিং ডিরেক্টরি', কাজের ডিরেক্টরি দেখাও, pwd, ব্যবহার করে আপনি দেখে নিয়েছিলেন রিডাইরেক্ট করা audio.cd.program.text ফাইলটা কোথায় সেভ করা হল। সেই '/home/dd/multimedia/audio' ডিরেক্টরি।

এবার যেই ওয়ার্কিং ডিরেক্টরি বলে একটা কিছু আছে, অমনি আমাদের পথ দেখানোর উপায়টাও তার সাপেক্ষে বদলে যাবে। প্রথমে ফাইলদুটোকে দেখে নিন, গোল চিহ্নে, audio.cd.program.text আর audio.filetypes.list, যারা আপনাকে কিছুতেই ব্যাশ শেলে টাইপ করার ট্যাবারাম পেতে দিচ্ছিল না দুটোরই শুরুতে ওই 'audio.' থাকায়। ফাইলদুটোর বজ্জাতি এখানেই শেষ নয়, ছবিতে ডিরেক্টরির ছকগুলো বানিয়েছিলাম আমীর সাইজের, তার মধ্যে এই বচ্চন সাইজের ফাইল, কী বিকট দেখাচ্ছে, কিন্তু, সরি, কিছু করার নেই, আজ তেরোই ডিসেম্বর, শনিবার, পরশু জয়েন করছি, এখন আর সময় নেই নতুন করে এতগুলো

পাতা ধরে ফাইলনাম বদলে আসার, এবং ভাঙার পক্ষে আমার আরো একটা ঠ্যাং এই মুহূর্তেই তৈরি আছে ঠিকই, হাত নয় কারণ লেখা যাবেনা, কিন্তু আর ছুটি নেই। ইন ফ্যাক্ট যা শুনছি, এতেই বোধহয় উইদাউট পে — তখন তো আপনাদের পাবনা। এবার ওই বারো হাত কাঁকুড়ের তেরো হাত বীচি ব্রান্ডের ফাইলদুটোকে দেখুন। এর অ্যাবসলিউট বা চূড়ান্ত ঠিকানাটা ভাবুন, শুরু হয়েছে একদম উপরের রুট ডিরেক্টরি স্ল্যাশ '/' থেকে, তারপর 'home', তারপর 'dd' মানে একজন ব্যবহারকারী, এইটা হল ইউজার ডিরেক্টরি, আগের হায়েরার্কির ছকের সঙ্গে মেলান, তার মধ্যে 'multimedia', তার মধ্যে 'audio' — এইখানে রয়েছেন 'audio.cd.program.text'।



তাহলে চূড়ান্ত পথনির্দেশটা মেলানো গেল '/home/dd/multimedia/audio/audio.cd.program.text'। এবার এটাকে আপেক্ষিক ভাবে ভাবুন। ধরুন, আগের দিন ফাইলটাকে সেভ করে বেরিয়ে গেছি। এবার পরের দিন লগ-ইন করলাম। মনে আছে সেই পুণ্যের কাজের কথা — পথহারা ব্যবহারকারীকে তার ঘর দেখায় লগ-ইন, পাঁচ নম্বর দিনে আমরা লগ-ইনের কাজে আলোচনা করেছি। এবার, আমি লগ-ইন কোথায় করেছি? আমার ইউজার ডিরেক্টরিতে, মানে '/home/dd'। ঢুকেই আমি '`pwd`' কমান্ডটা লিখে যদি এন্টার মারি, কমান্ড প্রম্পটে ফুটে উঠবে '`/home/dd`'। এবার এই '' ডিরেক্টরিতে আমার ঠিক সামনে কী আছে? তিনটে ডিরেক্টরি — documents, html, এবং multimedia। আমি যেতে চাই 'multimedia' ডিরেক্টরির মধ্যে 'audio' ডিরেক্টরিতে। এইজন্যে এবার আমি যে কমান্ডটা ব্যবহার করব, তার নাম 'chdir'।

আমি কমান্ড দেব '`chdir multimedia`'। এবার আবার '`pwd`' করলে ও দেখাবে '`/home/dd/multimedia`'। এবার ভাবুন তো, যার ঠিকানাটা '`/home/dd/multimedia`', সেখানে আমি পৌঁছে গেলাম শুধু '`chdir multimedia`' করে। এখানে আমি যদি '`chdir /home/dd/multimedia`' করতাম, তাহলেও একই কাজ হত, কিন্তু তার মানে তখন আমি চূড়ান্ত বা অ্যাবসলিউট পথ ব্যবহার করছি। আমার ওয়ার্কিং ডিরেক্টরির সাপেক্ষে রিলেটিভ পথ নয়। এখানে সরাসরি শুধু '`chdir multimedia`' করেই কাজ হল, কারণ ওই ডিরেক্টরিটা ঠিক আমার ওয়ার্কিং ডিরেক্টরিতেই বিরাজ করছে। এবার এই /home/dd/multimedia ডিরেক্টরিতে দাঁড়িয়ে আমরা সেই 'mv' কমান্ডটার অন্য কাজটাও করতে পারব, যা আগেরবার পারিনি, কারণ তখন আমরা এই পথনির্দেশ ব্যাপারটা জানতাম না। আগেরবার আমরা শুধু একটা ফাইলের নাম বদলেছিলাম, এবার একটা ফাইলকে নড়াব, তখন এটা পারিনি, কারণ, এটা করা মানেই ডিরেক্টরি কাঠামোর ভিতর নতুন ঠিকানাটা আমায় দিতে হবে। আমি এখন রয়েছি /home/dd/multimedia ডিরেক্টরিতে, এখানে আমার ওয়ার্কিং ডিরেক্টরিতে রয়েছে, ছবিতে দেখুন, audio, video, আর garbage নামে তিনটে ডিরেক্টরি। এই audio ডিরেক্টরির মধ্যে আছে সেই ঝামেলা পাকানো ফাইল audio.filetypes.list। এবার তাকে আমরা নড়িয়ে garbage ডিরেক্টরিতে নিয়ে যাব। আমরা কমান্ড দেব '`mv audio/audio.filetypes.list garbage/audio.filetypes.list`'। আমরা ব্যাবহার করলাম আপেক্ষিক বা রিলেটিভ পথ, কারণ এই ডিরেক্টরিতে আমার সামনেই রয়েছে audio আর garbage ডিরেক্টরিদুটো, তার একটার থেকে একটা ফাইল নিয়ে আমি অন্য ডিরেক্টরিতে রাখতে বললাম। এতক্ষণ আমরা '`ls`' কমান্ড দিয়েছি যে ডিরেক্টরির

ফাইল দেখছি সেই ডিরেক্টরিতে দাঁড়িয়ে, কারণ আমরা পথনির্দেশ জানতাম না, এবার আমরা এখানে দাঁড়িয়েই কমান্ড দেব — '`ls garbage`', এবং দেখতে পাব ফাইলটা ওই ডিরেক্টরিতে চলে গেছে। এই রিলেটিভ পথেই কাজ চলে গেল, কারণ, এই garbage ডিরেক্টরিটা আমার ওয়ার্কিং ডিরেক্টরিতেই আছে। যদি আমি /home/dd ডিরেক্টরিতে দাঁড়িয়ে থাকতাম, কমান্ড দিতাম '`ls multimedia/garbage`'। এখানেও হত রিলেটিভ পথ, একটু বড় রিলেটিভ পথ, কারন যে ডিরেক্টরিটা আমি দেখতে চাইছি, সেটা ওয়ার্কিং ডিরেক্টরিরই একটা সাবডিরেক্টরিতে আছে যার নাম multimedia। ধরুন, আমরা যদি এই পথের সম্পূর্ণ বাইরে, রুট ডিরেক্টরির নিচেই সবচেয়ে বাঁদিকের সাবডিরেক্টরিটায় থাকতাম, মানে '/bin' ডিরেক্টরিতে, তাহলে আমাদের ওই একই ফাইল দেখার জন্যে কমান্ড দিতে হত '`ls /home/dd/multimedia/garbage`'। এটা হত চূড়ান্ত বা অ্যাবসলিউট পথ।

৫।। ঐক্যবদ্ধ ফাইল সিস্টেম

আমাদের পথনির্দেশ জানা হল, কী করে একটা হায়েরার্কিকাল বা ক্রমানুসারী ডিরেক্টরি কাঠামোয় একটা ফাইলের বা ডিরেক্টরির ঠিকানা দিতে হয়। এবার আমরা যাব, এই ডিরেক্টরি কাঠামোর গ্নু-লিনাক্স আকার ঠিক কী হয় তার আলোচনায়। তার আগে, এই শেষ ছবিটায় ঠিক রুট ডিরেক্টরির উপরে প্রথম স্তরের সাবডিরেক্টরিগুলো খেয়াল করুন, /bin, /boot, /dev, /etc, /home, /lib, /sbin, /root, /opt, /proc, /mnt, /tmp, /usr, /var। এগুলো কিন্তু আনতাবড়ি নয়, আমার সুজে ৮.২ সিস্টেমের সত্যিকারের ডিরেক্টরি কাঠামোটাই দিয়েছি। এই ছবিটা একটু মন দিয়ে দেখে রাখুন, এরপর বারবার এটার কথা আসবে। '/' এবং তার পরেই তার থেকে প্রবাহিত এই ডিরেক্টরিগুলো — এই গোটাটা মিলিয়ে তৈরি হয় গ্নু-লিনাক্সের ইউনিফায়েড বা ঐক্যবদ্ধ ফাইল সিস্টেম। খেয়াল করুন, এটা একটা সিঙ্গল হায়েরার্কিকাল স্ট্রাকচার বা এক-উৎসের ক্রমানুসারী ডিরেক্টরি কাঠামো। একটা উৎস, রুট ডিরেক্টরি '/' — তার থেকে প্রবাহিত হয়েছে আর সমস্ত ডিরেক্টরি। আপনি যে ফাইল যে ডিরেক্টরিই ভাবুন না কেন তা এই মূল উৎস থেকে প্রবাহিত কোনো না কোনো ধারায়, বা কোনো না কোনো উপধারায় আছে।

উইন্ডোজ থেকে গ্নু-লিনাক্সে অভিবাসীরা প্রথম প্রথম তাদের হারিয়ে যাওয়া ড্রাইভ নিয়ে হাপিত্যেশ করেন, কোথায় গেলি C:\ মানে সি-ড্রাইভ, বা D:\ বা A:\ ইত্যাদি। এখানে তেমন কোনো ড্রাইভের গল্পই নেই। ওইরকম আলাদা আলাদা ড্রাইভ আছে বলে উইন্ডোজ হল একটা মাল্টিপল হায়েরার্কিকাল ডিরেক্টরি স্ট্রাকচার, বা বহু-উৎসের ক্রমানুসারী ডিরেক্টরি কাঠামো। সি-ড্রাইভ, ডি-ড্রাইভ, এ-ড্রাইভ, এই সবগুলোরই নিজের নিজের এক একটা রুট বা মূল ডিরেক্টরি আছে। যেখানটা নিছক C:\, এবার যেই আপনি উইন্ডোজ ডিরেক্টরিতে ঢুকলেন সেটা হয়ে গেল, C:\Windows\ — যদি আপনি এমএসডস-প্রম্পটে থাকেন তাহলে তো বটেই, উইন্ডোজ এক্সপ্লোরারে থাকলেও দেখবেন লোকেশন বারে, মানে একদম উপরের দিকে, মেনুবারের নিচেই সাদা জায়গাটায়, এই ঠিকানাটা ফুটে উঠছে, যেই আপনি বাঁদিকের প্যানেলে উইন্ডোজ ডিরেক্টরির উপর ক্লিক করলেন, এবং ডানদিকে ফুটে উঠল সেই ডিরেক্টরির মধ্যের সাবডিরেক্টরি এবং ফাইলগুলো। এরকম রুট ডিরেক্টরি উইনডোজে থাকে একটা করে, কিন্তু গ্নু-লিনাক্সে তথা যে কোনো ইউনিক্সে রুট ডিরেক্টরি হল এক এবং অদ্বিতীয় '/'। এইজন্যে গ্নু-লিনাক্স ফাইলসিস্টেমকে একটা ঐক্যবদ্ধ ফাইল সিস্টেম বলা হয়।

একটু বাদেই আমরা দেখব একটা মেশিনের কোন হার্ড ডিস্কের কোন কোন পার্টিশনকে এই ঐক্যবদ্ধ ফাইল সিস্টেমের ভিতরে নিয়ে আসা হবে সেটা নিজের ইচ্ছের উপর নির্ভর করে। একে টেকনিকাল পরিভাষায় আমরা বলি মাউন্ট করা। যে যে পার্টিশনে মাউন্ট করা হয় সেই সেই পার্টিশনের নিজের ফাইল সিস্টেম হায়েরার্কি এবার গোটা সিস্টেমের ফাইল সিস্টেম হায়েরার্কির একটা উপধারা হয়ে পড়ে। ধরুন, ওই সি-ড্রাইভ বা C:\ যে পার্টিশনে আছে তাকে মাউন্ট করা হয়েছে সেটা হল /mnt/windows/c/, এবার ঠিক সেই সেই ফাইল এবং ডিরেক্টরি সেই সেই সজ্জায় আমি পাব এই /mnt/windows/c/ ডিরেক্টরিতে যাদের যেভাবে আমি পেতাম C:\ ড্রাইভে। অর্থাৎ ওই সি-ড্রাইভের নিজের ফাইলসিস্টেম হায়েরার্কিটা এবার রুট ডিরেক্টরি '/'-এর ভিতর একটা উপধারা হয়ে গেল, মানে একটা সাবসেট হয়ে গেল ঐক্যবদ্ধ ফাইল সিস্টেমের সিঙ্গল হায়েরার্কির। এখন থেকে সি-ড্রাইভ যে ভৌত পার্টিশনে আছে সেই পার্টিশনে রাখা সমস্ত কিছুকে আমরা পাব /mnt/windows/c/, সেইজন্যে এই ডিরেক্টরিটা হল ওই পার্টিশনের মাউন্ট পয়েন্ট। এই ভাবে যে কোনো পার্টিশনেই মাউন্ট করা যায়, মানে তাকে নিজের ঐক্যবদ্ধ ফাইল সিস্টেমের মধ্যে ঢুকিয়ে আনা যায়। এমনকি সেই পার্টিশন যদি নেটওয়ার্কে সংযুক্ত অন্য কোনো মেশিনেও থাকে।

এমনকি একটা পার্টিশনকে দুটো সিস্টেম শেয়ারও করতে পারে। মানে দুজনেই ব্যবহার করতে পারে। কিছু কিছু ডিরেক্টরির বেলায় পারেনা, যেমন /boot, /etc, বা /var — এই ডিরেক্টরিগুলো যেখানে মাউন্ট করা সেই পার্টিশন দুটো সিস্টেম একই সাথে ব্যবহার করতে পারেনা। /etc ডিরেক্টরিতে থাকে সিস্টেমের সমস্ত কনফিগারেশন ফাইল, যেমন পাঁচ নম্বর দিনেই দেখেছেন আপনার সিস্টেমের প্রতিটি পাসওয়ার্ড থাকে এই ডিরেক্টরিতেই, সেটা অন্য সিস্টেমের সঙ্গে শেয়ার করবেন কী

করে? /boot ডিরেক্টরিতে থাকে আপনার সিস্টেমের কারনেল এবং তার খুঁটিনাটি, সেটাও সম্ভব না। একটু পরেই, লিনাক্স ফাইল সিস্টেম হায়েরার্কির সর্বজনগ্রাহ্য নিরিখ বা স্ট্যান্ডার্ডের আলোচনায় আমরা এগুলো ভালো করে জানতে পারব।

এই ঐক্যবদ্ধ ফাইল সিস্টেমের আর একটা গুরুত্বপূর্ণ জায়গা হল গ্নু-লিনাক্সে অত্যুন্নত মেমরি ব্যবহার ব্যবস্থা, যার অন্য নাম ক্যাশে। গোটা মেমরি ব্যবহারের প্রক্রিয়াটাকেই যা অনেকটা দ্রুততর করে তোলে। শূন্য এক আর দুই নম্বর দিনে বারবার আমাদের আলোচনায় এসেছে, কম্পিউটারে মূলত দুই রকমের মেমরির কথা, র‍্যাম আর রম। এই র‍্যামকে আবার দুইভাবে দেখেছি আমরা, ভৌত এবং ভৌতিক, ফিজিকাল এবং ভারচুয়াল। এই ভারচুয়াল মেমরি, সোয়াপফাইল, ক্যাশে — গোটা ব্যবস্থাটাই চলে একটা বিপুল পরিমাণ তথ্য অস্থায়ী ভাবে ভৌত ডিস্কে তুলে রাখার ভিত্তিতে। এই ব্যবস্থাটা এত ধারালো বলে গ্নু-লিনাক্সে মেমরির ব্যবহার চমৎকার। উইনডোজ ব্যবস্থায় র‍্যাম খুব বেশি বাড়ালে যেমন আদতে সিস্টেমের গতি কমেও যেতে পারে, যেটা ঘটে সেটা বাংলায় বললে এই যে সিপিইউ-র স্ক্রু ঢিলে হয়ে যায় ওই স্তূপ স্তূপ মেমরির পাতার মধ্যে পড়ে, কী খুঁজছে সেটাই হারিয়ে ফেলে। ফাইনালের আগের দিন রাত্তিরে ধরুন আপনাকে দেওয়া হল, কালকের কোশ্চেন আসতে পারে মাত্র এই একুশটা বই থেকে, পড়ে ফেলো বাবা। পরের দিন আপনি হলে গিয়ে, কুল বাংলার কোশ্চেনের উর্দুর উত্তর লিখে চলে এলেন।

গ্নু-লিনাক্স আসলে ঘুঘু জিনিষ, সে গোটাটা পড়েই না, টুকলি বানাতে থাকে। এবার, ঐক্যবদ্ধ যদি না হয় গোটা সিস্টেমটা, তাহলে, আবার সেই সমস্যা, কোথাকার টুকলি কোথায় যাবে, কোন মেমরি বাফার কোথায় লেখা হবে। এই জন্যে একটা গ্নু-লিনাক্স সিস্টেমে সঠিক শাট-ডাউনটা খুব জরুরি। হঠাৎ করে বিদ্যুৎ চলে গেলে বা অফ করে দিলে সমস্যা কিছু হতে দেখিনি আমি কখনো, বা শোনাও যায়না তেমন, গ্নু-লিনাক্সের ফাইল সিস্টেম এত জোরালো বা রোবাস্ট। কিন্তু সিস্টেমের উপর প্রচণ্ড চাপ পড়ে। অন্য অনেক উপায় আছে শাট-ডাউনের, ধীরে ধীরে আপনি সবগুলোই শিখে যাবেন। তবে একটা বেশ সহজ এবং নিরাপদ উপায় আছে, কন্ট্রোল-অল্ট-ডিলিট মারা, তারপর আঙুল উঁচিয়ে বসে থাকা, ঠিক নতুন করে বুট করে যাবার আগেই অফ করে দেওয়া। এটা গুই বা এক্স-উইনডোজে থাকলে হবেনা, সেখানে লগ-আউট করে তারপর কন্ট্রোল-অল্ট-ডিলিট মারতে হবে। এক্স-উইনডোজ থেকে বেরোনোর একটা শর্টকাট হল কন্ট্রোল-অল্ট-ব্যাকস্পেস, তবে সেটা ব্যবহার না-করাই ভালো।

এই ঐক্যবদ্ধ ফাইল সিস্টেমের ছবিটার /home ডিরেক্টরির মধ্যে দেখুন, আবার চারটে ইউজার ডিরেক্টরি, /home/atithi, /home/manu, /home/piu এবং /home/dd। পাঁচ নম্বর দিনে দেখার জন্যে তুলে দেওয়া /etc/passwd ফাইল থেকে মিলিয়ে নিন। আর, একটু মন দিয়ে খেয়াল করুন /root ডিরেক্টরিটা। এটা কিন্তু গোটা হায়েরার্কির চূড়ান্ত মা-ডিরেক্টরি মানে স্ল্যাশ / নয়। এটা হল সুপারইউজার বা রুট নামক ব্যবহারকারীর হোম ডিরেক্টরি। অন্য সব ইউজারের হোম ডিরেক্টরি হয় /home ডিরেক্টরির ভিতর। শুধু এর হোমটাই হয় অন্য জায়গায়। এই যে ডিরেক্টরি কাঠামোটা আমি এখানে দেখিয়েছি, এটা গ্নু-লিনাক্সের একটা বিশেষ ফ্লেভারের, যার নাম সুজে ৮.২, ডিস্ট্রিবিউশনটা সুজে, তার ৮.২ তম ভার্শন। এই ডিরেক্টরি কাঠামোটা, তার খুঁটিনাটি চেহারায়, এমনকি তার উপর উপর অবয়বেও, ফ্লেভার থেকে ফ্লেভারে বেশ বদলায়। আবার একটা একদেশতাও থাকে, গ্নু-লিনাক্স বলে নিজেকে ডাকতে গেলে কিছু ফাইল সিস্টেম হায়েরার্কি স্ট্যান্ডার্ড তাকে মানতে হয়। কিন্তু সেই ন্যূনতম জায়গাটুকুকে বাদ দিয়ে অনেক কনফিগারেশন বা আকারায়ন একটা থেকে অন্যটায় বদলায়। তাই, এই কাঠামোটা মুখস্থ না রেখে এর ধারণাটা মাথায় রাখার চেষ্টা করাই ভালো। এর কোনটার কী মানে, কেন থাকে, এইসবে আসছি — তার আগে এখানে একটা তালিকা দিয়ে রাখি, কত ফ্লেভার আছে আলাদা আলাদা গ্নু-লিনাক্সের, এটা আমি খুঁজে বার করিনি, ২৪ ঘন্টার হাইস্পিড কানেকশন না-থাকলে ওসব করা যায়না, একটা ভারি চমৎকার বই থেকে তুলে দিচ্ছি। বইটার নাম 'ফ্রি ফর অল — হাউ লিনাক্স অ্যান্ড দি ফ্রি সফটওয়ার মুভমেন্ট আন্ডারকাট দি হাইটেক টাইটানস', লেখকের নাম পিটার ওয়েইনার, http://www.wayner.org/books/ffa/ — বইটায় কোনোরকম প্রতিবন্ধ নেই, যে চাইবে তাকেই দেওয়া যাবে, আপনি নিজেও নামিয়ে নিতে পারেন, ১.৩ এমবি।

বইটা শুরু হচ্ছে নিরানব্বই-এর সেই কুখ্যাত গেটস-মামলা দিয়ে — মাইক্রোসফট গোটা সফটওয়ার ব্যবসায় মনোপলি তৈরি করে অন্য সবার ব্যবসা করার আয় করার উপায় বন্ধ করে দিচ্ছে কিনা। এই মামলায়, সে যে মনোপলি নয়, তার যথেষ্ট ভয়াল সব প্রতিদ্বন্দ্বী আছে এটা বোঝাতে গিয়ে গেটস-এর উকিল উল্লেখ করেছিল লিনাক্স এবং এফএসএফ-এর কথা। পিটার ওয়েইনার এই গোটা রসিকতাটার অভ্যন্তরস্থ বিকটতাটাকে চমৎকার দেখিয়েছেন — প্রশ্নটা এখানে টাকা, ক্যাশ, অন্য সবার টাকা আয় বন্ধ করে দেওয়ার জন্যে নোংরামি তুমি করছ কি করছ না, আর তাতে উদাহরণ দিচ্ছ লিনাক্স আর এফএসএফের যারা প্রথম থেকেই বিনা পয়সায় দিয়ে দিচ্ছে গোটাটা, এবং শুধু নিজেদের সফটওয়ারকে নিয়ন্ত্রণ করছে না তাই নয়, সফটওয়ার-প্রকৌশল-কম্পিউটার-চিন্তার গোটা প্রবাহটা যাতে কিছুতেই নিয়ন্ত্রণ করা না-যায় সেই ব্যবস্থা করার চেষ্টা করছে তাদের জিপিএল লাইসেন্স দিয়ে, যাকে অনেকেই মজা করে ডাকে 'কপিলেফট'। বইটা পড়ুন, সত্যিই ভালো লাগবে।

এই বইটা থেকেই তুলে দিই লিনাক্স ডিস্ট্রিবিউশনের তালিকাটা, এবং ইংরিজি অক্ষরেই দিচ্ছি, কারণ এর অনেকগুলোর নামই বাপের জন্মে শুনিনি আমি, আর অনেকগুলোই হল ইউরোপিয় নাম। একদম ল্যাটিন বর্ণমালার অনুক্রমে, Alzza Linux, Apokalypse, Armed Linux, Bad Penguin Linux, Bastille Linux, Best Linux (Finnish/Swedish), Bifrost, Black Cat Linux (Ukrainian/Russian), Caldera OpenLinux, CCLinux, Chinese Linux Extension, Complete Linux, Conectiva Linux (Brazilian), Debian GNU/Linux, Definite Linux, DemoLinux, DLD, DLite, DLX, DragonLinux, easyLinux, Enoch, Eridani Star System, Eonova Linux, e-smith server and gateway, Eurielec Linux (Spanish), eXecutive Linux, floppyfw, Floppix, Green Frog Linux, hal91, Hard Hat Linux, Immunix, Independence, Jurix, Kha0s Linux, KRUD, KSI Linux, Laetos, LEM, Linux Cyrillic Edition, LinuxGT, Linux-Kheops (French), Linux MLD (Japanese), LinuxOne OS, LinuxPPC, LinuxPPP (Mexican), Linux Pro Plus, Linux Router Project, LOAF, LSD, Mandrake, Mastodon, MicroLinux, MkLinux, muLinux, nanoLinux II, NoMad Linux, OpenClassroom, Peanut Linux, Plamo Linux, PLD, Project Ballantain, PROSA, QuadLinux, Red Hat, Rock Linux, RunOnCD, ShareTheNet, Skygate, Slackware, Small Linux, Stampede, Stataboware, Storm Linux, SuSE, Tomsrtbt, Trinux, TurboLinux, uClinux, Vine Linux, WinLinux 2000, Xdenu, XTeamLinux, and Yellow Dog Linux। ওয়েইনারও লিখেছেন, এটা সমাপ্ত কিনা জানিনা, আর আমি তো জানিই-না। তবে যেকটার নাম জানি তার মধ্যে মাংকি লিনাক্স ছাড়া সবকটাই আছে। তবে মাংকিটা নিজেই কোনো ডিস্ট্রিবিউশান না একটু বদলে নেওয়া তা জানিনা, যেমন নপিক্স (Knoppix) হল ডেবিয়ান ডিস্ট্রিবিউশানটাকে একটু বদলে নেওয়া। তবে ওয়েইনারের বইটা ২০০০-এ বেরোনো আর গ্নু-লিনাক্সের মানচিত্রটা এত দ্রুত বদলায়, কারণ গ্রহজোড়া কমিউনিটির অগণ্য মানুষের জ্যান্ত মনগুলো বদলায়, আর মহাভারতের ধর্ম-যখের সেই প্রশ্নোত্তর, মন তো সবচেয়ে দ্রুতগতি, সার্কিটের মধ্যে বিচরমান তথ্যবিদ্যুতের চেয়েও। এবার ফাইলসিস্টেম হায়েরার্কি স্ট্যান্ডার্ড, এখানে যা আমি জাস্ট ছুঁয়ে যাব, গোটা বিষয়টা খুবই বড় এবং বেশ ইন্টারেস্টিং। এর দুটো চমৎকার ডকুমেন্ট আছে, রাস্টি রাসেল আর ড্যানিয়েল কুইনলানের, 'ফাইলসিস্টেম হায়েরার্কি স্ট্যান্ডার্ড', অন্যটা বিন এনগুয়েনের, 'লিনাক্স ফাইলসিস্টেম হায়েরার্কি', পরেরটা লিনাক্স ডকুমেন্টেশন প্রোজেক্ট www.tldp.org থেকে পাবেন। আর ট্যানেনবম এবং সুমিতাভ দাস তো আছেই।

৬।। ফাইলসিস্টেমের মধ্যে আলাদা আলাদা ফাইলসিস্টেম

দুয়েকটা ধারণা আগে স্পষ্ট করে নেওয়া যাক। ফাইলকাঠামো বা ফাইলসিস্টেম বলতে আমরা বোঝাচ্ছি, একটা ডিস্কে বা পার্টিশনে রাখা ফাইলগুলোর হিশেব রাখার জন্যে অপারেটিং সিস্টেম যে তথ্য কাঠামো ব্যবহার করে এবং যা যা তরকিব নেয় তার গোটাটাকেই। মানে, এক কথায়, গোটা ডিস্কের ফাইলেরা কী ভাবে সংগঠিত। আবার একটা ডিস্কে বা পার্টিশনে কী ভাবে ফাইল রাখা হবে তারও নানা আলাদা আলাদা উপায় আছে। এই আলাদা আলাদা উপায়গুলোকেও আমরা ফাইলসিস্টেম বলে ডেকে থাকি। যেমন আমার দুটো হার্ড ডিস্কের সাতটা পার্টিশনে আছে চার রকমের ফাইলসিস্টেম, এক্সএফএস, রাইজারএফএস, সোয়াপ, আর উইন্ডোজ ফ্যাট৩২। কোথায় কোনটাকে পাব সেটা নিজের ইচ্ছেমতন, যাকে বলে মাউন্ট করা, পরে আসছি সেসবে। এবার দেখুন, এই ফাইল রাখার রাইজারএফএস, এক্সএফএস, ফ্যাট৩২ — এই যে আলাদা আলাদা রকম, আরো আছে, ইএক্সটি২ ইএক্সটি৩ ইত্যাদি — এদেরও ডাকছি আলাদা আলাদা ফাইলসিস্টেম বলে, আবার যখন একটা মেশিনের একটা সিস্টেমের গোটা ফাইল হায়েরার্কি কাঠামোটাকে ভাবছি, সেটাকেও ডাকছি ফাইলসিস্টেম বলে। আমার মেশিনে উইন্ডোজ ফ্যাট৩২ পার্টিশনদুটোকে পাওয়া যায় /mnt/windows/c এবং /mnt/windows/d ডিরেক্টরিতে, একটা রাইজারএফএস পার্টিশনকে পাওয়া যায় /mnt/slackware ডিরেক্টরিতে, অন্যটাকে /mnt/arkive ডিরেক্টরিতে, আগের সেকশনের ছবির সঙ্গে মিলিয়ে নিন। আর আমার মেশিনে মূল সুজে ৮.২ ফাইলসিস্টেমটা আছে তিনটে এক্সএফএস ফাইলসিস্টেম পার্টিশন মিলিয়ে। এই শেষ বাক্যটায় দেখুন, দুবার 'ফাইলসিস্টেম' শব্দটা ব্যবহার হয়েছে দুটো আলাদা অর্থে। সুজে ৮.২ ফাইলসিস্টেম বলতে বোঝাচ্ছি গোটা হায়েরার্কিকাল ডিরেক্টরি কাঠামোটাকে যার মধ্যে রয়েছে প্রতিটি পার্টিশনের প্রতিটি ফাইল, দ্বিতীয়বার ফাইলসিস্টেম বলতে বোঝাচ্ছি আলাদা আলাদা ধরনের পার্টিশনে আলাদা আলাদা রকমে ফাইল রাখার কায়দাকে।

একটা পার্টিশন বলতে আমরা বুঝি কাঁচা ভৌত হার্ড ডিস্কের একটা অংশকে। আর একটা ফাইলসিস্টেমকে বানিয়ে তোলা হয় এরকম এক বা একাধিক পার্টিশনকে মিলিয়ে। যেমন আমার সুজে অপারেটিং সিস্টেমের ফাইল সিস্টেম গড়ে উঠেছে সব কটা পার্টিশনকে মিলিয়ে। আবার উইন্ডোজ অপারেটিং সিস্টেমের কাছে গোটা ফাইলসিস্টেমটা হল শুধু দুটো পার্টিশনের, তাই উইন্ডোজ দেখায় আমার মোট হার্ড ডিস্ক স্পেস ৮০ জিবি-র জায়গায় মাত্র ১৬ জিবি, কারণ লিনাক্স পার্টিশনগুলোকে ও দেখতেই পায়না। এই পার্টিশনগুলোকে সরাসরি আমরা চিনি এদের ডিভাইস ফাইল দিয়ে। আর তাতে ফাইল সিস্টেম আলাদা করে তৈরি করে নিতে হয়। তাই ফাইল সিস্টেম হায়েরার্কি বোঝার আগে একটু ডিভাইস এবং পার্টিশনগুলোর সম্পর্কে জানা দরকার। এই মুহূর্তে এই পুরোটা একটু হিব্রু লাগছে হয়ত, কিন্তু আজকের আলোচনা শেষ হতে হতেই আর লাগবেনা।

পার্টিশন	সাইজ	ফাইলসিস্টেম টাইপ	মাউন্ট ভূমি
প্রথম হার্ডডিস্ক — ভৌত সাইজ চল্লিশ জিবি — /dev/hda			
/dev/hda1	8	Win95 Fat32	/mnt/windows/c
/dev/hda5	8 gb	Win95 Fat32	/mnt/windows/d
/dev/hda6	21.5 gb	Linux Reiserfs	/mnt/slackware
দ্বিতীয় হার্ডডিস্ক — ভৌত সাইজ চল্লিশ জিবি — /dev/hdb			
/dev/hdb1	110 mb	Linux XFS	/boot
/dev/hdb2	260 mb	Linux Swap	—
/dev/hdb3	12 gb	Linux XFS	/
/dev/hdb5	25 gb	Linux Reiserfs	/mnt/arkive

আমার মেশিনের হার্ড ডিস্কদুটোর পার্টিশনগুলোর একটা তালিকা এখানে দিচ্ছি। একটু মিলিয়ে নিই। তারপর আমরা এদের ভালো করে বোঝার চেষ্টা করব। আমার মেশিনে দুটো হার্ড ডিস্ক, /dev/hda আর /dev/hdb, আগেই বলেছি — এখন আমরা জানি এর মানে /dev নামের ডিরেক্টরিতে দুটো ডিভাইস ফাইল, যাদের নাম hda আর hdb। দুটোরই সাইজ চল্লিশ জিবি করে। এখানে তাদের পার্টিশনগুলোর তালিকাটা দিলাম। দুটোই চল্লিশ জিবির বা চল্লিশ গিগাবাইটের হার্ডডিস্ক, কিন্তু পাওয়া যায় মোটামুটি সাড়ে সাঁইত্রিশ জিবি করে, কারণ অন্যটুকু লেগে যায় ফাইলসিস্টেম থাকার কাঠামো বানাতে। এবার তালিকাটার দিকে তাকান, এর একদম বাঁদিকে আছে ডিভাইস ফাইলের নাম, যা দিয়ে সিস্টেম এই পার্টিশনটাকে চিনছে। তারপর তার সাইজ, জিবিতে মানে গিগাবাইটে আর এমবিতে মানে মেগাবাইটে। তিন নম্বর স্তম্ভে হল ফাইল সিস্টেমের নাম, চার ধরনের আছে মোট, আগেই বলেছি। এবং একদম ডানদিকে, মাউন্ট পয়েন্ট। ‘মাউন্ট পয়েন্ট’ কাকে বলে আপনি জানেন না এখনো, কিন্তু এই সাতটা মাউন্ট পয়েন্টকেই দেখুন, আপনি পরিস্কার চিনতে পারবেন এক একটা ডিরেক্টরির নামে, দেখুন, দেখানো আছে আগের সেকশনের ছবিটায়। তার মানে এইটাই, পরে আমরা আরো ভালো করে বুঝব — মাউন্ট করা হয় একটা ভৌত ডিভাইসকে, একটা পার্টিশনকে, এবং মাউন্ট করার ভূমিটা হল একটা ডিরেক্টরি।

যখন /dev/hdb5 পার্টিশনটা মাউন্ট করা আছে, তখন আমরা /mnt/arkive ডিরেক্টরিতে ঢুকে সেখানে ls মারলে তার ফাইলগুলো দেখতে পাব — আমার সমস্ত ডাউনলোড, সমস্ত সফটওয়্যার, সমস্ত এভিআই ফাইলগুলো ওখানে রাখা — বেশ লোভনীয় ডিরেক্টরি। জিএলটির গোটা সিডিটা বানানো একটা ওয়েবপেজের মত করে, সেই সিডির মোট জিনিষপত্তরও ওখানেই রাখা linux.books নামের একটা ডিরেক্টরিতে। কিন্তু। কিন্তু যখন মাউন্ট করা নেই, আপনি chdir করে / থেকে /mnt থেকে /mnt/arkive ডিরেক্টরিতে যান, এবার ls মারুন। ফক্কা। কিছুই নেই। কারণ, ওই ভৌত উপাদান, /dev/hdb5 পার্টিশনটা, যাকে মাউন্ট করা হয় ওই ডিরেক্টরিতে, সে তখন ফাইলসিস্টেমের বাইরে, তাকে মাউন্ট করা হয়নি।

দেখুন তো, এবার কি একটু শিকারের গন্ধ নাকে আসছে? মনে হচ্ছে প্রায় পেড়ে ফেলেছেন, আর কয়েক থাবার দূরত্বে মাত্র? যদি তা নাও হয়, এই অদ্ভুত লাগাটা একদমই অপরিচয় জনিত। চেনা কাউকে দিয়ে নিজের মেশিনে ইনস্টল করে নিন গ্নু-লিনাক্স, হাতের কাছে খোঁজাখুঁজি করে একদমই কাউকে না-পেলে সবচেয়ে কাছের লাগ বা জিএলটি-তে যোগাযোগ করুন, এমনকি চেষ্টা করে নিজেও করে ফেলতে পারবেন। আমাদের জিএলটি-র দুজন নিজে নিজেই করেছে প্রথমবারই। যদিও টেলিফোন করতে হয়েছে কয়েকবার, মাঝপথে নানা অসুবিধেয়। এবার ওই ইনস্টল করা সিস্টেমে একটু সময় কাটাতে থাকবেন, প্রথম প্রথম মনে হবে সবই একটা ল্যাবিরিন্থ-এর মত, গোলোকধাঁধা, তারপর, হঠাৎ একসময় সিস্টেমের যুক্তিগঠনটা আপনার মাথায় বসে যেতে শুরু করবেন, দেখবেন, কুচি কুচি ম্যুরালের মত একটা ছকে বসে যেতে শুরু করেছে আপনা থেকেই। এই বসে যাওয়ার প্রক্রিয়াটা ত্বরান্বিত করার চেষ্টায় আসুন এবার একটু ডিভাইস, পার্টিশন, ডিস্ক, মাউন্ট এগুলোকে আর একটু ভালো করে জানার চেষ্টা করা যাক।

একটা ফাইলসিস্টেমের একদম গোড়ার কয়েকটা উপাদান হল সুপারব্লক, আইনোড, ডেটা ব্লক, ডিরেক্টরি ব্লক, এবং ইনডিরেকশন ব্লক। যেগুলো এবার একটু ধরে ধরে বুঝতে হবে আমাদের। প্রাথমিক সংজ্ঞাটা জেনে রাখা যাক। সুপারব্লকে থাকে গোটা ফাইলসিস্টেম মানে ওই পার্টিশনের গোটা ডিরেক্টরি ব্যবস্থা এবং ফাইলব্যবস্থা সম্পর্কে তথ্য। তার গঠন, তার আকার, এইসব। ঠিক কী কী তথ্য এই সুপারব্লকে থাকবে, এবং ঠিক কী ভাবে থাকবে এটা বদলায় ফাইলসিস্টেম থেকে ফাইলসিস্টেমে। ধরুন, পার্টিশনটা রাইজারএফএস হলে যেভাবে থাকবে এটা, এবং যা থাকবে সেই সুপারব্লকে, এক্সএফএস হলে সেভাবে থাকবেনা, এবং সেই তথ্যও থাকবেনা। ভিফ্যাট হলে আবার আর এক রকম। ইত্যাদি। আইনোডে থাকে একটা ফাইল সম্পর্কে যাবতীয় তথ্য, শুধু ফাইলটার নাম ছাড়া। কারণ, ফাইলসিস্টেমের কাছে নামটার কোনো মানে নেই, মানে আছে আপনার মানে ব্যবহারকারীর কাছে। ডিরেক্টরি ফাইলে লেখা থাকে ফাইলের নাম এবং আইনোড নম্বর, আগেই বলেছি। যাতে আপনি কোনো ফাইলকে টানাহেঁচড়া করছেন যখন, তার নামটা লিখছেন, ডাইরেক্টরি ফাইলে তার আইনোড নম্বরটা পড়ে সিস্টেম জেনে যেতে পারে কোন ফাইলটার কথা আপনি বলছেন। ফাইলের তথ্যটা লেখা আছে কোন কোন ডেটা ব্লকে সেটাও লেখা থাকে আইনোডে। আইনোডে মাত্র কয়েকটা ডেটা ব্লকের নম্বর লেখারই জায়গা থাকে, তার চেয়েও বেশি ডেটা ব্লকের যদি দরকার পড়ে ফাইলের মোট তথ্যটা লিখতে, তাহলে আরো আরো ডেটা ব্লকের পয়েন্টার ফাইলের সঙ্গে যোগ করা হয় একটা গতিশীল বা ডায়নামিক রকমে। এই গতিশীল ভাবে কাজে লাগানো ডেটা ব্লকগুলোকেই বলে ইনডিরেক্ট ব্লক, যাদের

নম্বর ঠিকানা ইত্যাদি লেখা থাকে ফাইলসিস্টেমের ইনডিরেকশন ব্লকে। একটু ভজোকটো লাগছে হয়ত, কেটে যাবে, বাস্তব একটা ফাইলসিস্টেমকে ধরে আলোচনা শুরু হলেই।

এই শেষ প্যারাটায় গোটাটা জুড়েই যে ফাইলসিস্টেমের কথা বলা হয়েছে সেটা এক একটা পার্টিশনে এক একটা রকমের ফাইলব্যবস্থা যে ভাবে থাকে, তার বিষয়ে। এই রকমের এক বা একাধিক পার্টিশনকে মিলিয়ে তৈরি হয় একটা পূর্ণাঙ্গ ডিরেক্টরি সিস্টেমের ফাইলব্যবস্থা। একটা ফাইলব্যবস্থা একটা পার্টিশনে গ্নু-লিনাক্সে প্রায় থাকেই না, সেটা গ্নু-লিনাক্স নিরাপত্তা এবং কাজের ধারণার সঙ্গেই মেলেনা। এবার এর সঙ্গে আরো আরো অপারেটিং সিস্টেম যোগ হলে তো সেটা আর ভৌত রকমেই সম্ভব না। স্ল্যাকওয়্যারের, অন্তত বুট করার জন্যে এবং একান্ত নিজস্ব কনফিগারেশন ফাইলের জন্যে, একটা সম্পূর্ণ নিজস্ব পার্টিশন লাগবেই, ডেটার পার্টিশনগুলো সে যদি শেয়ারও করে সুজের সঙ্গে। আবার উইন্ডোজ তো গ্নু-লিনাক্স পার্টিশনকে দেখতেই পাবেনা, তার মানে, উইন্ডোজ থাকলে তার নিজের পার্টিশন লাগবেই, যদি দেখতেও পেত, তাহলেও ওই বুট করার জন্যে এবং নিজের সিস্টেম কনফিগারেশন ফাইল রাখার জন্যে একটা পার্টিশন লাগতই। তার মানে ধরুন যে কটা অপারেটিং সিস্টেম আপনার আছে, অন্তত সেই কটা পার্টিশন আপনার লাগবেই।

যখন আমার শুধু উইন্ডোজ ছিল তখনই, সেই আট জিবির হার্ড ডিস্কে ছিল তিনটে পার্টিশন, তার অবশ্য একটা রক্তমাংসের কারণ ছিল, যা গ্নু-লিনাক্সে অনুপস্থিত, সেটা হল ভাইরাস। মাঝে মাঝেই শুরু হত ভাইরাসের ভুতের নেত্য, তখন আর কি, বাপ বাপ বলে একটা পার্টিশন পুরো ফরম্যাট করে দাও, অন্যটায় ফাইল এনে। আবার যদি নিজের ভালোবাসা মাস্টার বুট রেকর্ডে বা এমবিআর-এ ছড়িয়ে দেওয়ার মত রোমান্টিক ভাইরাস হয় তাহলে তো সবগুলোই ফরমাট করো, মানে কম্ম কাবাড়, গোটা হার্ডডিস্কের প্রতি কুচি তথ্য উপড়ে ফেলো। এরকম আবার একবারই হয়েছিল, তার আগের ফাইলগুলোর প্রিন্ট-আউট ছিল, মধ্যের তেইশদিনের কাজ চলে গেছিল ভাইরাসদের অন্ত্রে। মাঝে মাঝে মনে হয়ন, কোনো ক্রিয়াই তো অবশেষহীন হয়না, আর কম্পিউটারে লেখা মানেও তো একটা ভৌত ক্রিয়া, সেই ফাইলগুলো, তার তথ্যগুলো, গেল কোথায়, কোন শূন্যতায়? এটাই কি পরম ব্রহ্মে লীন হয়ে যাওয়া, যেখানে আছে, কিন্তু আর ফিরিয়ে আনা সম্ভব না।

এই যে দেখুন, দুবার বললাম 'ফরম্যাট' করার কথা, এখানে বলেছি তথ্যশূণ্য করে দেওয়া অর্থে, আমরা সবাই করি, একটা কাঁচা হার্ড ডিস্ক কিনে এনে সেটাকে সরাসরি ব্যবহার করা যায়না। তাকে ফাইল রাখার, তথ্য রাখার, পরে সেই তথ্য ব্যবহার করার মত করে একটা আকার দিতে হয়, মানে একটা ফাইলসিস্টেমের কঙ্কাল বানাতে হয়। শিখিয়ে পড়িয়ে নিতে হয়, সদ্য দলে আসা রাজনৈতিক কর্মীকে যেমন নেতা হওয়ার আগে, অনেক যত্নে কেয়ারে তিরষ্কারে, আগে সভ্য এবং চোর বানিয়ে তুলতে হয়। কোথায় কোন জিনিষের কী মানে এটা সে নিজে থেকেই যাতে বুঝে যেতে পারে। পুরোনো মানে দিয়ে যাতে আর না-বোঝে। গ্নু-লিনাক্সে এটা করা হয় মেক-ফাইলসিস্টেম (mkfs) দিয়ে। এর পুরোটারই একটা ছোট্ট অংশ হল আগে ভৌত হার্ড ডিস্কটাকে ইলেকট্রনিক তথ্যের থেকে মুক্ত করা।

৭।। ডিভাইস, ডিভাইস-ফাইল, ডিভাইস-নাম

আমরা জানি গ্নু-লিনাক্সে সব ডিভাইসই এক একটা ফাইল যারা /dev ডিরেক্টরিতে বা /dev ডিরেক্টরির পেটের মধ্যে আরো আরো সাবডিরেক্টরিতে আছে। যত দিন বেড়েছে, এই /dev ডিরেক্টরির আকার এবং জটিলতাও বেড়েছে, কারণ প্রকৌশল বদলেছে, আরো আরো নতুন ধরনের ডিভাইস আবিষ্কার হয়েছে, তাদের নিয়ে আসা হয়েছে ব্যবহারে, অপারেটিং সিস্টেমের মধ্যে। একটা ডিভাইস খোলা, ব্যবহার করা বন্ধ করা মানে, ওই ডিভাইসের ফাইলটাকে খোলা, পড়া, লেখা, বন্ধ করা। ডিভাইস পড়তে বললেই কী করে করতে হবে, এবং কী করতে হবে, কোথায় কোন ড্রাইভারকে কী ভাবে কাজে লাগাতে হবে, এটা কারনেল জানে, সেটা জানে বলেই সে কারনেল। দুই নম্বর দিনের আলোচনাগুলো মনে করুন। আজকের সেকশন ৪-এর শেষ ছবিটায় দেখানো বিভিন্ন সিস্টেম ডিরেক্টরিগুলোর আলোচনায় আমরা আসছি একটু পরেই, গ্নু-লিনাক্স ফাইলসিস্টেম হায়েরার্কির আলোচনায়, কোন ডিরেক্টরিতে কী থাকে এবং কেন থাকে গ্নু-লিনাক্সে তার চালু প্রথায়। এখানে কয়েকটা ডিভাইস ফাইলের সূত্রে /dev ডিরেক্টরির কথা একটু বলে নেওয়া যাক। এই ডিভাইস ফাইলগুলো একটা গড় উদাহরণ, এক একটা সিস্টেমে এর থেকে আলাদা হতেই পারে, হয়েই থাকে, আমরা ছকটা বোঝার চেষ্টা করছি।

/dev ডিরেক্টরিতে যে ফাইলগুলো থাকে তাদের মধ্যে আছে হার্ড ডিস্ক — /dev/hd, আমরা আগেই দেখেছি, দুটো হার্ড ডিস্ক থাকলে হয় /dev/hda এবং /dev/hdb। ধরুন, উইনডোজ হলে, যদি দুটো হার্ড ডিস্কেই একটা করে পার্টিশন থাকে তাহলে হত C:\ এবং D:\। আবার দুটো হার্ড ডিস্কেই দুটো করে পার্টিশন থাকলে হত C:\ D:\ E:\ F:\ ইত্যাদি। /dev/hda এবং /dev/hdb দুটোই আইডিই হার্ড ডিস্ক। আগেই তো বলেছি, আইডিই মানে ইন্টিগ্রেটেড ডিভাইস ইলেকট্রনিক্স, এক ধরনের ডিস্ক ড্রাইভ যাতে ডিভাইসের ড্রাইভারটা দেওয়া থাকে ড্রাইভের মধ্যেই, আলাদা করে অ্যাডাপ্টার কার্ড আর লাগাতে হয়না। এর থেকে উন্নত এবং তাই দামী হার্ড ডিস্ক হল স্কাসি, স্মল কম্পিউটার সিস্টেম ইন্টারফেস। আইডিই হার্ড ডিস্কের থেকে অনেক দ্রুত গতিতে তথ্য পড়া এবং লেখা যায় স্কাসি ড্রাইভে। একটা বিশেষ ধরনের ব্যাবস্থা এই স্কাসি। এতে যে শুধু হার্ড ডিস্ক হয় তা নয়, অন্য পেরিফেরালও হয়। স্কাসি হার্ড ডিস্ক প্রথমটার নাম হয় /dev/sda, দ্বিতীয়টার নাম হয় /dev/sdb। এবার আইডিই বা

স্কাসি দুরকম হার্ড ডিস্কেই একাধিক করে পার্টিশন থাকতে পারে, /dev/hdb1 মানে দ্বিতীয় আইডিই হার্ড ডিস্কের এক নম্বর পার্টিশন, /dev/sda3 মানে প্রথম স্কাসি হার্ড ডিস্কের তিন নম্বর পার্টিশন, ইত্যাদি। আইডিই সিডি ড্রাইভ হলে তার ডিভাইস ফাইল হয় /dev/sr0, আর একটা থাকলে /dev/sr1, এখানে শূন্য মানে প্রাইমারি, এক মানে স্লেভ। এদের লিংক বা সংযোগ করা থাকে /dev/cdrom0, /dev/cdrom1 ইত্যাদি নানা ফাইল দিয়ে। সংযোগ করা বলতে কী বোঝায় সেকথায় আমরা পরে আসব। স্কাসি সিডি ড্রাইভ হলে তখন নাম হয় /dev/scd। একাধিক থাকলে /dev/scd0, /dev/scd1। শুধু হার্ড ডিস্ক নয়, সিডি রম, মেমরি এরকম অনেক কিছুই স্কাসি হতে পারে। উইনডোজে যাকে কমিউনিকেশন পোর্ট কম-ওয়ান (COM1) বলে, গ্নু-লিনাক্স তাকে ডাকে /dev/ttys0 বলে। মানে প্রথম সিরিয়াল পোর্ট। সচরাচর এতে মোডেম লাগানো থাকে। তখন /dev/ttys0 আর /dev/modem পরস্পর সংযুক্ত করা থাকে। একটা বললেই অন্যটাকে বোঝায়। /dev/modem থেকে আমরা তথ্য পড়ছি মানে /dev/ttys0 পোর্টে যুক্ত মোডেম থেকে কোনো মেল বা ওয়েবপেজ নামাচ্ছি মেশিনে। ইউএসবি (USB — **U**niversal-**S**erial-**B**us) ডিভাইসগুলোর ফাইলগুলো থাকে /dev/usb ডিরেক্টরিতে। ইউনিভার্সাল সিরিয়াল বাস হল একটা উন্নত বাস-ব্যবস্থা, বাস মানে তো আমরা জানি, পেরিফেরালগুলোর সঙ্গে যোগাযোগপথ, মোটামুটি '৯৮-এর পর থেকে ব্যবহৃত হচ্ছে। পিএসবাইটু (PS/2) পোর্টের ডিভাইস ফাইল হয় /dev/psaux। এই পিএসটু এসেছে আইবিএম-এর একটা পিসি মডেলের নাম থেকে, পিসিটা খুব একটা চলেনি, অন্য ধরনের একটা নির্মাণের কারণে, কিন্তু নামটা রয়ে গেছে। পিএসবাইটু পোর্টে মাউস, মোডেম ইত্যাদি ব্যবহার হয়। ফ্লপি ড্রাইভ সচরাচর হয় /dev/fd0। /dev/dsp মানে তো আগেই বলেছি, সাউন্ড কার্ড বা প্রথম সাউন্ড ডিভাইস। /dev/lp0 মানে প্রিন্টার পোর্ট বা এলপিটি-ওয়ান (LPT1)। এই যে নামকরণগুলো লিখলাম, ডিরেক্টরিতে দেখবেন, কখনো কখনো তার একটা যৌক্তিক তালমিল পাওয়া যায় বটে, কিন্তু বেশিরভাগ কেসেই বেশ ইচ্ছেমাফিক মর্জিমাফিক।

এই ডিভাইসগুলোর মধ্যে আবার ব্লক ডিভাইস আর ক্যারেকটার ডিভাইস এই দুই ভাগ আছে, আমরা আগেই বলেছি, তার সঙ্গে মিলিয়ে ফাইলগুলোও ব্লক ডিভাইস ফাইল আর ক্যারেকটার ডিভাইস ফাইল। যেমন, ফ্লপি, হার্ড ডিস্ক, সিডি ড্রাইভ — এগুলো ব্লক ডিভাইস, আবার সিরিয়াল পোর্ট, মাউস, প্যারালাল প্রিন্টার পোর্ট — এগুলো ক্যারেকটার ডিভাইস। ব্লক ডিভাইসে তথ্য লেখা এবং পড়াটা হয় ব্লকের এককে। একটা প্রোগ্রাম যখন একটা ফাইলকে লিখতে চায় একটা ব্লক ডিভাইসে, সে আবেদন জানায় কারনেলের কাছে, কারনেল তখন ডিভাইস ড্রাইভারকে কাজে লাগিয়ে এক এক তাল তথ্য, মানে এক এক ব্লক করে লিখতে থাকে ডিস্কের শরীরে থাকা ওই ফাইলে, প্রতিটি ব্লকের একটা নিজের নম্বর আছে, দুই নম্বর আর এক নম্বর দিনের আলোচনা থেকে মনে করুন। আবার ডিস্ক থেকে একটা ফাইল পড়ছি যখন, প্রথমে খোঁজা হয় বাফার ক্যাশে, মানে সেই তুলে রাখা অস্থায়ী স্মৃতি, সেখানে সাম্প্রতিকতম অতীতে ব্যবহৃত তথ্যের মধ্যে পাওয়া যায় কিনা। যদি মিলল তো গুড, আর ডিস্ক নাড়াচাড়া করতে হলনা, অনেকটা সময় বেঁচে গেল, একটু আগে গ্নু-লিনাক্স ব্যবস্থার মেমরি ব্যবহারের যে বাড়তি ওস্তাদিটার কথা বলছিলাম।

এবার আমাদের শিখে যাওয়া একটা কমান্ড একটু ব্যবহার করা যাক /dev ডিরেক্টরির দুচারটে ফাইলের উপর। আমরা /dev ডিরেক্টরিতে গিয়ে যদি '`ls -l -R`' মারি, তাহলে একটা বিরাট সময় ধরে অজস্র ফাইলের লং লিস্টিং বা দীর্ঘ তালিকা স্ক্রিন জুড়ে দেখিয়ে যাবে সিস্টেম, যেখানে একটা ফাইলের প্রচুর খুঁটিনাটি ফুটিয়ে তুলবে সিস্টেম। আমরা কিন্তু এখানে '`ls -lR`' কমান্ডও মারতে পারতাম, '`ls`' কমান্ড তার অপশানগুলোকে এভাবে জুড়ে দিতে দেয়, যদিও সব কমান্ড দেয়না। কোনটা দেয়, কোনটা দেয়না এটা আপনাকে নিজেকে শিখতে হবে। গ্নু-লিনাক্স কমিউনিটির অজস্র মানুষ মিলে শুধু আপনি সেখান থেকে পড়ে শিখবেন বলে ওই বিপুল পরিমাণ টেক্সট লিখে রেখেছে, যা আপনার সিস্টেমে এমনিতেই দেওয়া থাকে। এখানে এই '`l`' অপশানটা বলে ফাইলের সমস্ত প্রয়োজনীয় খুঁটিনাটিকে লং বা দীর্ঘ আকারে ফুটিয়ে তুলতে, আর '`R`' অপশানটা বলে ওই ডিরেক্টরির মধ্যে যে সাবডিরেক্টরিগুলো আছে তাদের মধ্যের ফাইলগুলোকেও দেখাতে। এই শেষহীন তালিকাটাকে রিডাইরেক্ট করে একটা ফাইল বানাতেও জানি আমরা। সেই ভাবে তৈরি করা ফাইল থেকে কয়েকটা লাইন তুলে দিই, আদতে ফাইলটায় লাইনের সংখ্যা এগারো হাজারের মত। আসলে যা যা ঘটতে পারে সব সম্ভাবনাই তৈরি থাকে, যেমন আটখানা হার্ড ডিস্কের প্রতিটায় যাতে একত্রিশটা অব্দি পার্টিশন বানানো যায়, hda1 থেকে hda31, hda থেকে hdh, আরো বারোখানা হার্ড ডিস্কে যাতে পনেরোটা অব্দি পার্টিশন বানানো যায়, hdi1 থেকে hdi15, hdi থেকে hdt, সেই ব্যবস্থাও করে রাখা আছে আগে থেকেই। এবং এটা সব রকম ডিভাইসের জন্যেই। এই ডিভাইসাড়ম্বরের একটা যুক্তি নিশ্চয়ই আছে, আমি জানিনা, ভালো করে কম্পিউটার জানে এমন কারো কাছে জিগেশ করতে হবে।

```
lrwxrwxrwx  1 root  root       10 2003-12-05 14:53 modem -> /dev/ttyS0
crw-rw----  1 root  uucp   4,  64 2003-03-14 18:37 ttyS0
crw-rw--w-  1 root  lp     6,   0 2003-03-14 18:37 lp0
brw-rw----  1 root  disk   3,   0 2003-03-14 18:37 hda
brw-rw----  1 root  disk   3,   1 2003-03-14 18:37 hda1
brw-rw----  1 root  disk   3,  64 2003-03-14 18:37 hdb
```

```
brw-rw----  1 root  disk   3,  65 2003-03-14 18:37 hdb1
lrwxrwxrwx  1 root  root       10 2003-12-05 07:09 mouse -> /dev/psaux
crw-rw----  1 root  root   10,  1 2003-12-17 11:17 psaux
lrwxrwxrwx  1 root  root        8 2003-12-05 07:38 cdrom -> /dev/sr1
lrwxrwxrwx  1 root  root        8 2003-12-05 07:38 cdwri -> /dev/sr0
brw-r--r--  1 root  disk   11,  0 2003-03-14 18:37 sr0
brw-r--r--  1 root  disk   11,  1 2003-03-14 18:37 sr1
brw-r--r--  1 root  disk   11,  0 2003-03-14 18:37 scd0
brw-r--r--  1 root  disk   11,  1 2003-03-14 18:37 scd1
```

এবার দেখুন তো কয়েকটা ফাইলের এই লং লিস্টিং থেকে আপনি কিছু আন্দাজ করতে পারছেন কিনা? ঠিক এই আগের দুতিনটে প্যারাগ্রাফে যা আমরা বলে এলাম, সেইগুলোকে? প্রতিটি ফাইলের নামের আগেই সিস্টেমের কাছে তার অ্যাবসলিউট বা চূড়ান্ত ঠিকানা '/dev/' যোগ করে নিন, কারণ, এই 'ls' কমান্ডটা তো আমরা দিয়েছি '/dev' ডিরেক্টরিতে দাঁড়িয়েই।শুধু একটা জিনিষ মনে রাখুন, লাইনের একদম গোড়ায় যে 'b' বা 'c' — তার মানে ব্লক ডিভাইস বা ক্যারেকটার ডিভাইস, আর 'l' মানে লিংক, ওই যে সংযোগের কথা বললাম, পরে আরো ভালো করে জানব। এবার একদম ডানদিকের ডিভাইসের নামের সঙ্গে মেলাতে পারছেন? চারটে ডিভাইসের নামের পরে দেখুন '->' চিহ্ন দিয়ে অন্য একটা ডিভাইস দেখানো, এর মানে মুখ আর মুখোশ। যেমন '/dev/psaux' ডিভাইসের মুখোশ হল '/dev/mouse'। কোনো প্রোগ্রাম '/dev/mouse' বললে সিস্টেম আসলে '/dev/psaux' বুঝবে। আমরা '`cat /dev/mouse`' করে মাউসের নড়াচড়ার টেক্সট সংকেত স্ক্রিনে দেখেছিলাম, মনে পড়ছে? এই মেশিনে মাউস ডিভাইসের ফাইলটা '/dev/psaux' হল কারণ এটা পিএসবাইটু মাউস। এই ডিভাইস ফাইলগুলো এবং বাস্তব ভৌত উপাদানগুলো সঠিক অর্থেই তুল্যমূল্য, আমরা আগেই, বিভিন্ন ডিভাইস ফাইলগুলো নিয়ে মজা করার সুত্রে দেখিয়েছি, কী ভাবে একদম ভৌত উপাদানগুলোর মতই এই ডিভাইসগুলোয় তথ্য লেখা যায়, বা এখান থেকে তথ্য পড়া যায় — মাউস-চলনের ফরমুলা বা কারনেলের কণ্ঠস্বর তারই প্রমাণ।

ডিভাইসদের একটা ভাগ যেমন ব্লক আর ক্যারেকটার, তেমনি আর একটা ভাগ হল মেজর আর মাইনর নাম্বার, মুখ্য আর গৌণ সংখ্যা। মেজর নাম্বার বা মুখ্য সংখ্যা দিয়ে বোঝায় ওই ধরনের ডিভাইসের বর্গ, আর মাইনর নাম্বার বা গৌণ সংখ্যা দিয়ে বোঝায় ডিভাইসটার নিজস্ব বর্গ। ডিভাইসটা মাস্টার না স্লেভ, বা কত নম্বর পার্টিশন এগুলো বোঝায় এই গৌণ সংখ্যা দিয়ে। ফাইলের ওই দীর্ঘ তালিকাটায় দেখুন পাঁচ নম্বর স্তম্ভটা হল মেজর আর মাইনর নাম্বারের। যেমন '`hda1`' ডিভাইসের সংখ্যাদুটো হল '`3, 1`', আবার '`hdb1`' ডিভাইসের সংখ্যাদুটো '`3, 65`'। একই ডিভাইস কন্ট্রোলার তাদের দুজনকেই নিয়ন্ত্রণ করে বলে এদের দুজনেরই মেজর নাম্বার সমান। এই জায়গাটা নিয়ে আমার আর না-এগোনোই ভালো, আমি নিজেও এইরকম ভাসাভাসা ভাবেই জানি। ভালো করে জানতে চাইলে কোলকাতা লাগে আমাদের মেইলিং লিস্ট আছে কী করতে? আপনি কত চান? পড়ে আর কূল পাবেন না। জিসিসিতে math.h বলে একটা হেডার ফাইল আমি, নিজের বুদ্ধির দোষেই, ভালো করে ম্যানুয়াল পড়িনি, কিছুতেই কাজে লাগাতে পারছিলাম না, লিস্টে জানানোর দু-ঘন্টার মধ্যে গোটা পাঁচেক মেল এসেছিল, কৌশিক ঘোষ বলে একজন ধরে ধরে কমান্ড তুলে তুলে আমায় বুঝিয়ে দেয়, এইসব ছেলেরা যা ভালো জানে সব — সে আর বলার না।

কিন্তু এখানে একটা সমস্যা তৈরি হচ্ছে, এই ডিভাইস ফাইলগুলো নিয়ে কথা বলার সময় দেখছিলাম, কিছু কথা বোঝানো যাচ্ছেনা, মানবসমাজ ও সভ্যতার আদিমতম সমস্যা মালিকানা এবং অধিকার নিয়ে কিছু কথা না-বলে। কম্পিউটারের ফাইল ব্রহ্মাণ্ডের ভূগোল নিয়ে কথা বলা শেষ করার আগে অব্দি তার এথিকস নিয়ে কিছু শুরু করব না, ঠিক করেছিলাম, কিন্তু সেটা দেখছি হচ্ছেনা, ফাইলের সম্পত্তিসম্পর্ক নিয়ে একটা সেকশন নামিয়ে নেওয়া যাক। আসলে এই বইটা লেখার আগে এই বইটা তো কোনোদিন লিখিনি, কিছু কিছু জায়গা ঘেঁটে যাচ্ছে। যাকগে, আপনারাও তো এটা পড়ার আগে কোনোদিন এটা পড়েনইনি, পড়ার পরেও পড়বেন বলে মনে হয়না। কিন্তু, আজকের আলোচনা এই অব্দিই থাক। এর পরের আলোচনা পরের দিন।

glt-mad@ilug-cal.org

MADHYAMGRAM

আবার আমাদের প্ল্যান বদলালো। আজকে শেষ হলনা ফাইলসিস্টেম। আসলে লেখার আগে যেভাবে ভেবেছিলাম, লেখার সময়ে দেখছি তার থেকে বদলে যাচ্ছে গোটাটা। অনেক কথা মাঝপথে বলে নিতে হচ্ছে, যা অন্যভাবে বলার কথা ভেবেছিলাম। এর পরের দিনটাও গোটাটাই এই ফাইলসিস্টেমে চলে যাবে বলেই মনে হচ্ছে। আমরা শুরু করব ফাইলের অনুমতি মালিকানা নিরাপত্তা নিয়ে। তারপর যাব একটা ভৌত পার্টিশনে কী করে ফাইল সিস্টেম লেখা থাকে সেই কথায়। তারপর শেষ করব গ্নু-লিনাক্স ফাইল সিস্টেম হায়রার্কি দিয়ে। এখন লেখাটাও এগোচ্ছে অত্যন্ত বিরক্তিকর রকমের শ্লথগতিতে। আরো এখন তো কলেজ চলছে। রোজ কলেজ করে আর কতটুকু সময়ই বা পাচ্ছি লেখার? দেখুন না, আজ একুশে ডিসেম্বর, তাও তো এটায় যা যা

রাখার প্ল্যান ছিল প্রথমে তার অনেকটাই রাখতে পারলাম না। পরের দিন করতে হবে। ফাইলসিস্টেম হায়েরার্কি শেষ হয়ে যাবে আজ এরকমই ভেবেছিলাম। নিজেকেও অনেক পড়তে হচ্ছে। জিএলটি ইশকুলের ক্লাসের চেয়ে এখানে পরিশ্রম অনেক বেশি হচ্ছে, বলার সময়ে যা চট করে বলে চলে যাওয়া যায়, নিজেরও খেয়াল পড়েনা, দুটো চারটে করে ডিটেইলস-এর ফাঁক রয়ে যাচ্ছে, ঠিক নিজে যে ভাবে ভাবি সেভাবেই তো বলি। কিন্তু এখানে লিখতে গিয়ে অনেক খুচরো পাপের প্রায়শ্চিত্য করতে হচ্ছে। আমার মাথায় নয় করার সময় এই কৌতূহলটা আসেনি, অন্য আর একজনের তো আসতেই পারে, তার তো কাজে আসবে না তখন টেক্সটটা। আর আপনিও সক্রিয় হয়ে উঠুন গ্নু-লিনাক্সে, এবং তার কমান্ড মোডেই, উইনডোজ-এর মত গুই দিয়ে নয়, সেটাই তো এই বইটা চায়। শেখানোটা বড় নয়, সেই কাজে আমি খুব দড়ও নই, নিজেই তো শিখছি একটু একটু করে, বড় কথা আপনার চিন্তা আর কৌতূহলকে উত্তেজিত করা। সেটা পারছি কিনা কে জানে? লিখি তো সবসময়েই, কিন্তু এই লেখাটা আমার কাছেও খুব ভিন্ন — নিজের এলাকার বাইরে একটা টেকনিকাল আন্দোলনের গতিটাকে হাজির করা আমারই মত অটেকনিকাল মানুষদের কাছে। যারা ঠিক মত টেকনিকাল রকমে কম্পিউটার শেখেন তাদের জন্যে তো এই বই নয়।

সংকলন ও রচনা : মধ্যমগ্রাম জিএলটি-র (glt-mad@ilug-cal.org) তরফে : ত্রিদিব সেনগুপ্ত

গ্নু-লিনাক্স ইশকুল

জিএলটি মধ্যমগ্রাম
glt-mad@ilug-cal.org

ভাবা ছিল যে ফাইলসিস্টেম নিয়ে আলোচনা ছ নম্বর দিনেই শেষ হবে, কিন্তু হলনা। এখনো কিছু জিনিষ বলার আছে এই ফাইলসিস্টেম নিয়ে। ফাইলের অনুমতি মালিকানার সম্পত্তি-সম্পর্ক নিয়ে কথা বলার পর আমরা যাব একটা ভৌত পার্টিশনে ফাইল সিস্টেম গড়ে ওঠার ব্যবস্থাগুলো নিয়ে। তারপর যাব ফাইলসিস্টেম নিয়ে আমাদের শেষ প্রসঙ্গ, গ্নু-লিনাক্স কাঠামোয় ফাইলসিস্টেম হায়েরার্কি স্ট্যান্ডার্ড।

।। দিন সাত।।

১।। ফাইলের সম্পত্তিসম্পর্ক

এক নম্বর দিনে যখন লিখছিলাম, তখন কিছুতেই বুঝতে পারছিলাম অনভ্যস্ত একজন কাউকে আমি সুপারইউজার আর ইউজার-এর ধারণাটা কী করে বোঝাব। ওই যে ছিলনা, কারনেল স্তরে খাপ খুলতে পারে কেবল রুট বা সুপারভাইজর বা সুপারইউজার, আর ইউজার কাজ করে একদম উপরের প্রয়োগ বা অ্যাপ্লিকেশন সফটওয়ারের স্তরে। দেখুন, সেই অবস্থাটা আমরা বেশ পেরিয়ে এসেছি, অন্তত আমার তো তাই মনে হচ্ছে। এখানে একটা বিচ্ছিরি সময়গত সমস্যা আছে, চিরকাল চিঠি লেখার সময়ে সেটা আমার মনে পড়ে যেত, একটা কিম্ভুতকিমাকার লাগত, চিঠিগুলোই কেমন গুলিয়ে যেত। সেটা লেখকের মুহূর্ত আর পাঠকের মুহূর্ত। লেখার মুহূর্ত আর পড়ার মুহূর্তের মধ্যে দিন মাস বছর আর কিলোমিটারের দূরত্ব। চিন্তার আবেগের অনুভূতির মুহূর্তের দূরত্ব। এই যে আমি লিখলাম ‘বেশ’, আর আপনি হয়ত যোজন যোজন দূরত্ব ঠেলে ভাবছেন, একে ‘বেশ’ বলে, ছ্যা। কিন্তু আমি তো আপনার মুহূর্তে বাঁচছিনা, তাই কিছু করার নেই।

এখন আমরা জানি ইউজার বলতে কী বোঝায়, যেমন এই যে সিস্টেমের উদাহরণটা আমি দিয়ে চলেছি, আমার সিস্টেমটাকে একটু বদলে নিয়ে, এখানে ব্যবহারকারী বা ইউজার হল atithi, manu, piu এবং dd। তাদের বাড়ি বা হোম ডিরেক্টরি হল যথাক্রমে /home/atithi, /home/manu, /home/piu এবং /home/dd। মিলিয়ে নিন, পাঁচ নম্বর দিনে দেওয়া /etc/passwd ফাইল থেকে। আর সুপারইউজার হল root, তার হোম ডিরেক্টরি হল /root। আগেই তো বলেছি, রুট শব্দটা দুটো অর্থেই ব্যবহার হয় গ্নু-লিনাক্সে। রুট ডিরেক্টরি মানে /, আর রুট ইউজারের হোম ডিরেক্টরি মানে /root। এবং এই সুপারইউজার হল একটা গ্নু-লিনাক্স সিস্টেমে সর্বশক্তিমান, যা যা কাজ করা যায় তার প্রত্যেকটাতেই তার অপ্রতিহত অধিকার। সব কাজ সে পারেনা, যেমন একজন সুপারইউজার একটা ডিরেক্টরি ফাইলের ভিতরটা দেখতে পায়না, বলেছি আমরা, একজন সুপারইউজার আপনার সেই প্রবন্ধের কোনো প্রিন্টআউট নিতে পারেনা যা আপনি সামনের বছর লিখবেন, এই রকম।

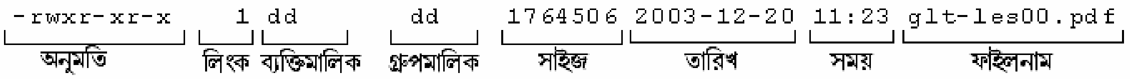
কিন্তু সুপারইজার নয় অপ্রতিহতগতি, ইউজাররা তাই বলে কি বানের জলে ভেসে এসেছে নাকি? ইউজারেরও বেশ কিছু অধিকার আছে গোটা সিস্টেমে, নইলে আপনি কাজ করবেন কী করে? কারণ, আগেই তো বললাম, সবসময়েই মেশিন ব্যবহার করবেন ইউজার হয়ে। ইউজারদের এই সাধ্য এবং অধিকারের মধ্যে আবার নানারকম প্রকারভেদ আছে, নানা গ্রুপ আছে ইউজারদের। একজন ব্যবহারকারী যতগুলো খুশি গ্রুপেরই সভ্য হতে পারে, প্রাথমিকভাবে কোনো সিস্টেমে কোনো নতুন ইউজার তৈরি হওয়ামাত্র সে অন্তত গ্রুপের একটা মেম্বার হয়ই, তার নিজের নামে যে গ্রুপের নাম, এবং একজনই সভ্য সেই গ্রুপের। এটা সচরাচর সিস্টেমের ডিফল্ট সেটিং, তারপরে যা খুশি তো বদলে নেওয়াই যায়।

৮.১।। ফাইলের মালিকানা এবং অনুমতি

যখনই কোনো ফাইল তৈরি হয় একটা সিস্টেমে, যে কোনো ফাইল, সেই ফাইলের একটা ব্যক্তি মালিক এবং একটা গ্রুপ মালিক ঘোষিত হয়ে যায়, সিস্টেম সেটা দেগে দেয় ফাইলের খুঁটিনাটিতে। যেকোনো ফাইলের উপর ‘`ls -al`’ মেরে যেটা দেখা যায়। যেমন ‘/home/dd’ ডিরেক্টরির ভিতর একটা সাবডিরেক্টরিতে একটা ‘`ls -al`’ কমান্ডের ফলাফল এখানে তুলে দেওয়া যাক, ‘/dev’ ডিরেক্টরির উপর ‘`ls -al`’ কমান্ডের ফলাফলের একটা অংশ আগের সেকশনেই আমরা দিয়েছিলাম।

```
-rwxr-xr-x    1 dd     dd      1764506 2003-12-20 11:23 glt-les00.pdf
-rwxr-xr-x    1 dd     dd       227703 2003-12-20 11:23 glt-les01.pdf
-rwxr-xr-x    1 dd     dd       202997 2003-12-20 11:23 glt-les02.pdf
-rw-r--r--    1 root   root        294 2003-12-20 11:24 ld.so.conf
-rw-------    1 root   root        637 2003-12-20 11:24 lilo.conf
-rw-r--r--    1 root   root      13227 2003-12-20 11:24 modules.conf
```

এখানে ‘ls -al’ যে ছকটা ব্যবহার করেছে স্ক্রিনে তালিকা ফুটিয়ে তোলার সেটা এবার বোঝা যাক। প্রথমে তালিকার প্রথম মানে ‘glt-les00.pdf’ ফাইলটাকেই ধরুন। প্রথমে মালিকানা দেখানো হয়েছে ‘-rwxr-xr-x’ অংশটায়, তারপর ‘1’ অংশটায় দেখানো হয়েছে এর লিংক বা সংযোগের সংখ্যা কটা, তারপর ‘dd’ হল এর ব্যক্তি মালিক মানে ইউজার ‘dd’, তারপরে ‘dd’ মানে এর গ্রুপ মালিক মানে গ্রুপ ‘dd’। এরপর এর সাইজ বাইটে, ‘1764506’, একে মেগাবাইটে বদলে নিলে সাইজটা দাঁড়াচ্ছে ১.৬৮ এমবি। তারপর ফাইলটাকে শেষ বদলানোর তারিখ এবং সময়, ‘2003-12-20’ এবং ‘11:23’। শেষে ফাইলের নাম ‘glt-les00.pdf’। এর মধ্যে অন্যগুলো তো জলের মত বুঝতে পারছেন, শুধু অনুমতি আর লিংকে এখনো একটু ব্যথা আছে, সেটা সমাধান হয়ে যাবে এই সেকশনেই।



এই অনুমতি অংশ বলতে আমরা যেটা দেখিয়েছি সেখানে ছকটা হল দশটা হাইফেনের, এইরকম ‘----------’, এর একদম বাঁদিকেরটা হল ফাইলের চরিত্র বোঝাতে, সেটা ডিরেক্টরি হলে ‘d’, রেগুলার ফাইল হলে এই হাইফেনটা শূন্য থাকবে। ক্যারেকটার ডিভাইস হলে এটা হবে ‘c’, ব্লক ডিভাইস হলে এটা হবে ‘b’, লিংক হলে ‘l’ — এই তিনটেকে আগের সেকশনের তালিকা থেকে মিলিয়ে নিন। এবার তারপরে রইল নটা, সেটা আসলে তিনটে ড্যাশের তিনটে গ্রুপ, ‘--- --- ---’, এখানে আসতে পারে তিনটে অক্ষর, ‘r’, ‘w’ এবং ‘x’। ‘r’ মানে রিড বা পড়ার অনুমতি, ‘w’ মানে রাইট বা লেখার মানে বদলানোর অনুমতি, ‘x’ মানে একজিকিউট বা চালানোর অনুমতি। প্রথম তিনটে হাইফেন হল ব্যবহারকারীর নিজের, দ্বিতীয় তিনটে হাইফেন হল তার গ্রুপের, সেখানে যদি অন্য কেউ থাকে তার কী কী করার অনুমতি আছে, আর শেষ তিনটে হাইফেন হল অন্যদের জন্যে, মানে যারা ওই ব্যবহারকারী নয়, এমনকি তার গ্রুপেরও সভ্য নয়। এইখানে একটা অক্ষরও নেই, এর মানে, এটা যদি কোনো ফাইলের অনুমতি তালিকা হয় তো সেই ফাইলটা লিটারালি শিবঠাকুরের আপনদেশের ফাইল, সর্বনেশে এর আইনকানুন, ফাইলটার যে মালিক সে নিজেও ফাইলটা পড়তে বদলাতে বা চালাতে পারবেনা, শুধু ফাইলমালিকের পূর্ণ অধিকারে স্ক্রিনে ফাইলটার নাম আর অনুমতির তালিকার দিকে নির্নিমেষ চোখে চেয়ে থাকতে পারবে, তার উপরে গ্নু-লিনাক্সে কোনো বারণ নেই। সদ্য বেরোনো কারনেল ভার্শন ২.৬ অব্দি অন্তত নেই। ওই চেয়েই থাকতে হবে তাকে, যতক্ষণ না সে ‘chmod’ বলে একটা কমান্ড দিয়ে ফাইলটার এইসব বেলেল্লাপনাকে শায়েস্তা করছে, সেই কথায় আসছি। তবে সেই কমান্ড না-জেনে নিজে নিজে এইরকম একটা ফাইলের মালিক হওয়ার কোনো উপায় আমি জানিনা। ফাইলের এই বিচিত্র আচরণ ঘটানো যায় একমাত্র ওই কমান্ডটা দিয়েই। আর একটা ফাইলের অনুমতি তালিকাটা যদি এই রকমের হয়, ‘rwxrwxrwx’, তার মানে, এই ফাইলটার উপর ব্যবহারকারীর নিজের, তার গ্রুপের এবং অন্যদের প্রত্যেকের ফাইল পড়ার, লেখার এবং চালানোর প্রতিটি অনুমতিই আছে।

এবার দেখুন তো, আপনি একটু আগের তালিকাটা থেকে, বা আগের সেকশনের তালিকাটা থেকে, এই অনুমতি ব্যাপারটা পরিষ্কার বুঝে যাবেন, হাইফেন থাকা মানে অনুমতিটা নেই, অক্ষর থাকা মানে আছে, প্রথম তিনটে হাইফেন নিজের, পরের তিনটে গ্রুপের, শেষ তিনটে অন্যের। এবার কাজটা এখানেই শেষ নয়, গোটা সিস্টেমে যেখানে যেখানে পারেন, এই অনুমতির চক্করটা বোঝার চেষ্টা করুন। শুধু একটা কথা বলি, ডিরেক্টরির বেলায় পড়ার বা লেখার বা চালানোর অনুমতি মানেটা একটু বদলে যায়, স্বাভাবিকভাবেই, ডিরেক্টরি তো প্রোগ্রাম না, কোনো গাড়িও না যে চালালেই হল। ধরুন একটা ডিরেক্টরি a, তার মধ্যে একটা ফাইল আছে b। অন্যদের বেলায় এই a ডিরেক্টরিটা পড়ার অনুমতি নেই, কিন্তু লেখার বা এক্সিকিউট করার অনুমতি আছে। এই অবস্থায় অন্যেরা সেই ডিরেক্টরিতে ls করে b ফাইলটা যে আছে এটা দেখতে পারবেনা, কিন্তু যদি জানে যে আছে তাহলে সেই b ফাইলটা পড়তে পারবে, বা নতুন করে একটা c ফাইল লিখতেও পারবে সেই a ডিরেক্টরিতে, মজার কথা লেখার পরেও সেটা ls করে দেখতে

পারবেনা। আবার a ডিরেক্টরিতে পড়ার বা এক্সিকিউট করার অনুমতি আছে, কিন্তু লেখার অনুমতি নেই, এই অবস্থায় অন্য কেউ ডিরেক্টরিতে ls করে কী আছে দেখতে পারবে, b ফাইলটা পড়তেও পারবে, কিন্তু ওই নতুন c ফাইলটা লিখতে পারবে না ওই ডিরেক্টরিতে। আর, পড়ার বা লেখার অনুমতি আছে কিন্তু এক্সিকিউট করার অনুমতি যদি না থাকে a ডিরেক্টরিতে, তখন ls করে ফাইল দেখা, বা b ফাইল পড়া, বা c ফাইল লেখা, সবই বন্ধ।

৮.২।। ফাইলের অনুমতি বদলানো

এবার কথা হল, যে অবস্থায় আছে, সেখান থেকে, দরকার মত আপনি একটা ফাইলের অনুমতি বদলাবেন কী করে। অনুমতি বদলানোর জন্যে দরকার ওই ফাইলটার উপর আপনার মালিকানা। আর সুপারইউজার বা রুট যে কোনো ফাইলের মালিকানা যে কোনো সময় বদলে দিতে পারে। এই বদলানোর দুটো প্রক্রিয়া আছে, একটা হল অক্ষর দিয়ে, অন্যটা হল সংখ্যা দিয়ে।

অক্ষর দিয়ে বদলানো মানে কোনো একটা ফাইলের মালিকের সাপেক্ষে তিন ধরনের ব্যবহারকারীর জন্যে তিনটে অক্ষর, 'u' মানে সে ইউজার নিজে, 'g' মানে তার গ্রুপের যে কোনো ব্যবহারকারী, আর 'o' মানে আদারস বা অন্য আর সবাই, আর 'a' মানে সবাই, যার মধ্যে 'u', 'g', বা 'o' সকলেই পড়ে। আর অনুমতি তো আমরা আগেই জানি — 'r' মানে রিড বা পড়ার অনুমতি, 'w' মানে রাইট বা লেখার অনুমতি, 'x' মানে একজিকিউট বা চালানোর অনুমতি। আর অনুমতি দেওয়া মানে '+', অনুমতি তুলে নেওয়া মানে '-'। এবার ধরুন, আমি essay.all.text নামের একটা প্রবন্ধ সবার পাঠে ধন্য হোক এটা আমি চাইছি, তখন আমি কমান্ড দেব 'chmod'।

কমান্ড দেব, 'chmod a+r essay.all.text'। যদি চাই এটা হবে একটা সামাজিক নির্মাণ, স্তালিনিস্ট রকমের একটা প্রবন্ধের অ্যাসেম্বলি লাইন, সবাই লিখতে পারুক, সাম্যবাদী গনতান্ত্রিক লেখক সংঘ, কমান্ড দেব 'chmod a+w essay.all.text'। ডিরেক্টরির বেলাতেও এই একই ভাবে কাজ করতে হবে। শুধু ডিরেক্টরির বেলায় একটা বাড়তি অপশান দেওয়া যাবে '-R'। এর মানে রিকার্সিভ বা পারম্পরিক। ধরুন আমি কমান্ড দিলাম 'chmod -R g+x my.directory', এর মানে শুধু যে 'my.directory' ডিরেক্টরিটা আমার গ্রুপের সকলের কাছে এক্সিকিউটেবল হয়ে গেল তাই নয়, এর মধ্যের যাবতীয় ফাইলরা, ছানা সাবডিরেক্টরিরা, তাদের ছানারা, বংশপরম্পরায় সকলেই এক্সিকিউটেবল হয়ে গেল আমার গ্রুপের কাছে।

এই অক্ষর দিয়ে অনুমতি বদলানোটা আমার বেশ জটিল লাগে, এত কিছু মনে রাখতে হয়। এর চেয়ে সহজ লাগে সংখ্যা দিয়ে বদলানোটা। সেখানে '4' মানে পড়া, '2' মানে লেখা, '1' মানে এক্সিকিউশন। এবার পড়া আর লেখার অনুমতি মানে '4' আর '2', মানে '6', পড়া আর চালানোর অনুমতি মানে '4' আর '1', মানে '5', লেখা আর চালানোর অনুমতি মানে '2' আর '1', মানে '3', আর সবকটার অনুমতি মানে '4' আর '2' আর '1' মানে '7'। কোনো অনুমতি নেই মানে '0'। এবার 'chmod' কমান্ডটার সঙ্গে একটা তিন অঙ্কের সংখ্যা আসতে পারে, তার প্রথমটা হল ইউজারের নিজের, পরেরটা হল তার গ্রুপের, আর শেষেরটা হল আদারস বা অন্যের। এবার ধরুন আমি সবাইকে সব অনুমতি দিতে চাইছি 'somefile' ফাইলটার উপর, তার মানে কমান্ডটা হবে, তিনটে ঘরের জন্যেই '7'। মানে, 'chmod 777 somefile'। আমি নিজের জন্যে পড়ার লেখার আর চালানোর অনুমতি চাইছি, গ্রুপের জন্যে চালানোর আর পড়ার, অন্যের জন্যে চালানোর, তার মানে কমান্ডটা দাঁড়াবে, 'chmod 751 somefile'। এবার করে করে দেখুন, কী হয়। যত করে দেখবেন, মাথায় বসিয়ে ফেলতে পারবেন ছকটা, তত আপনি গ্নু-লিনাক্স শেখার কাছাকাছি যাচ্ছেন।

খুব ঝঞ্ঝাটের লাগছে, না? লাগবে না, বরং কৃতজ্ঞ লাগবে, যখন মনে হবে, এটাই সেই নিরাপত্তা, যা দিয়ে আপনার গ্নু-লিনাক্স সিস্টেমের ভাইরাস-অনাক্রমণীয়তাটা গড়ে উঠেছে, সেই ইমিউনিটি। নিজে ভাবার চেষ্টা করুন তো, ভাইরাস তো এক ধরনের প্রোগ্রাম, যারা সিস্টেমে ঢুকে, সিস্টেম ব্যবহারকারীদের ইচ্ছে এবং প্রয়োজনের বাইরে গিয়ে সিস্টেমের ফাইল বদলাতে থাকে। এবার ভাবুন তো, কেন বলছি, যে এই মালিকানা আর অনুমতির ছকটাই নিরাপত্তা ব্যবস্থা। ধরুন এটা এই জিএলটি ইশকুল পাঠমালার টাস্ক রইল, কোনো খুঁটিনাটি না, কিছু না, জাস্ট এই যতটুকু জানেন এর ভিত্তিতেই ভেবে ওঠার চেষ্টা করুন তো কেন এটাকেই নিরাপত্তা বলছি?

ও, একটা কথা, এই যে কোনো মালিকানা এবং অনুমতির ছকে রুট বা সুপারইউজার কিন্তু পড়েনা, তাই রুট নিজে যদি সিস্টেমের বেগড়বাই ঘটাতে চায়, সে নিজেই যদি একটা ভাইরাসকে চালু করে, রুট ইউজার বা সুপারইউজার হয়ে,

তাহলে কিন্তু সে ভাইরাস তখন ভগবান, হল বলরাম স্কন্ধে। তবে, রুট হয়ে অত ভাইরাস চালানোর পরিশ্রমের দরকার কী, সিধে তো ফাইলসিস্টেমটাই উড়িয়ে দিতে পারে সে, একদম গোড়া থেকে, আগেই বলেছি।

৮.৩।। সিস্টেম এবং ইউজার

এবার, প্রত্যেক ইউজারের আলাদা আলাদা অনুমতি এবং মালিকানার এই গোটা কাঠামোটা সিস্টেম মাথায় রাখে কী করে? — /etc ডিরেক্টরির কয়েকটা ফাইলকে দিয়ে। তার কিছুটা কিন্তু আপনি ইতিমধ্যেই জানেন, পাঁচ নম্বর দিনের আলোচনায় /etc/passwd আর /etc/shadow ফাইলদুটোর কথা মনে করুন। মনে করুন, লগ-ইন কিভাবে এদের পড়ে। গোটা সিস্টেমের মধ্যে একমাত্র এই /etc/passwd ফাইলটাতেই থাকে প্রতিটি নথীবদ্ধ ইউজারের খুঁটিনাটি। কিছু কিছু ডিস্ট্রিবিউশনে, বিশেষ করে সাম্বা বলে একটা সফটওয়ারের কার্যধারায়, এর কিছু ব্যত্যয় ঘটে, তবে তা নিয়ে এখন মাথা না-ঘামালেও চলবে। সাতটা ফিল্ডে বাঁধা এর লাইন নিয়ে আমাদের আলোচনা মনে করুন, এর মধ্যে একটা লাইন আমরা তুলে নিয়েছিলাম, ‘`piu:x:503:100:Smita Bhadra:/home/piu:/bin/bash`’ — মনে আছে? এর মধ্যে, তৃতীয় আর চতুর্থ জায়গাটা অবস্থানটা আমরা বুঝতে পারিনি, ওই সংখ্যা দুটো, ইউজার আইডি আর গ্রুপ আইডি, ব্যক্তি-পরিচিতি আর গ্রুপ পরিচিতি। এখন বুঝতে পারছেন? এ দুটো হল দুটো সংখ্যা, যাদের দিয়ে সিস্টেম চেনে ‘piu’ নামক ইউজার আর তার গ্রুপকে, যাদের আমরা চিনি ইউজার ‘piu’ আর তার গ্রুপ ‘Users’। এখানে দেখুন নম্বরদুটো আলাদা, এটা গ্নু-লিনাক্সের অভ্যস্ত জায়গা থেকে একটু আলাদা, কারণ এখানে সুজেতে ‘ইউজারস’ বলে এই গ্রুপটা তৈরি করে সেখানে সাধারণ ব্যবহারকারীদের রাখা হয়। বললে ও অন্যভাবে, মানে প্রত্যেকের জন্যেই একটা ব্যক্তিগত গ্রুপ তৈরি করে, রাখতেই পারে। যেমন দেখুন ওই একই ফাইলের একই লাইনে, শুধু এটা এবার স্ল্যাকওয়ার থেকে, এখানে কিন্তু নম্বর দুটো এক, দুটোই ‘piu’। `piu:x:1004:1004:Smita Bhadra:/home/piu:/bin/bash` — এইখানে দেখুন ইউআইডি আর জিআইডি দুটোই এক। এবং রেডহ্যাট সিস্টেমেও, ডিফল্টে এভাবেই হয়, আলাদা করে গ্রুপ বিষয়ে কোনো অপশান না দিয়ে দিলে। সরাসরি ‘useradd’ বলে একটা কমান্ড দিয়ে রুট কোনো নতুন ইউজারকে নথীবদ্ধ বা রেজিস্টার করাতে পারে সিস্টেমে। এই ‘useradd’ কমান্ডটার ম্যানুয়াল পেজ বা ইনফো পেজ পড়ে দেখুন, /etc/passwd ফাইলের লাইনে একজন ইউজারের ওই সাতটা জায়গা বা ফিল্ডের প্রত্যেকটাকেই কিরকম আলাদা আলাদা ভাবে বদলে নেওয়া যায় রেজিস্টার করার সময়েই। তার ইউজারনেম, তার পাসওয়ার্ড, তার গ্রুপ সংক্রান্ত পছন্দ, তার পুরো নাম, তার হোম বা বাড়ি কোথায় হবে, তার শেল কী হবে। যেই রেজিস্টার্ড হয়ে গেল একজন ইউজার, তার নাম, ধাম, গোত্র, বাড়িতে পুকুর আছে কিনা, পুকুরে ঘটি ডোবে কিনা, ইত্যাদি চিত্রগুপ্ত লিখে ফেলল /etc/passwd ফাইলে, তার নাগরিকত্ব জানা হয়ে গেল সিস্টেমের, তার আগে অব্দি সিস্টেমের কাছে তার কোনো অস্তিত্ব নেই। একবার রেজিস্টার করার বা নথীবদ্ধ হয়ে যাওয়ার পরেও যদি একজন ব্যবহারকারীর খুঁটিনাটি বদলাতে চান, তার জন্যে কমান্ড আছে ‘usermod’। আবার একবার বলে দেব ম্যান পেজ পড়ার কথা? এখন থেকে, যখনই কোনো নতুন কমান্ড উল্লেখ হবে, ধরেই নেবেন তার সঙ্গে এই ধ্রুবপদটা লাগানোই আছে, ‘ম্যান পড়ার নেশা করো ম্যান, খেজুরগাছে হাঁড়ি বাঁধো ম্যান’ (কার্টসি দোহার)।

অনেক টেক্সটেই দেখেছি, রাশি রাশি কমান্ডের ব্যবহারবিধি পরপর দিয়ে যেতে, আমিও এখানে দিতেই পারতাম, কিন্তু আমি নিজে শেখার সময়ও তো দেখছি, নিজেই তো শিখছি, এই টেক্সটটার চেয়ে এক মার্জিন এগিয়ে, নিজেরই মনে হয়, এত বোগাস আর বোরিং এইগুলো লিখে দেওয়া, রাশিরাশি গুষ্টির ওই রাশিরাশিতর পিন্ডি, ও ছাই মনেও রাখা যায়না। একমাত্র মনে থাকে ম্যানপেজ পড়ে পড়ে, করে করে, দেখে দেখে। গ্নু-লিনাক্স সিস্টেম আপনাকে যা খুশি করার মজাটা যেমন দেয়, তেমনি তার জন্যে নিজেকে বানিয়ে নেওয়া লাগে। কেন এত বারবার বিরক্তিকর রকমে এই ম্যান পড়ার কথা বলছি এটা বুঝতে পারবেন একটা গ্নু-লিনাক্স সিস্টেমে নিজের মজায় কাজ করতে চাইলেই — ইট টেকস এ ওরিড ম্যান টু সিং এ ওরিড সং।

এবার আসে চিত্রদার দ্বিতীয় ফাইল, সেটাকেও চিনি আমরা, তার ছায়া-ছবি দেখেছি আমরা পাঁচ নম্বর দিনেই, /etc/shadow ফাইল। আগেই বলেছি, আগে প্রবেশ-সংকেত বা পাসওয়ার্ডটা এই /etc/passwd ফাইলটাতেই থাকত, পরে এই /etc/shadow ফাইলটা সিস্টেমে চালু হল একটা বিশেষ প্রয়োজন থেকে। /etc/passwd ফাইলটা নিয়ে সমস্যা এই যে সেটা যে কারুকেই যে কোনো সময়ে পড়তে হবে, জনপাঠযোগ্য বা ওয়ার্ল্ড-রিডেবল হতে হবে, তার

মানে এর ফাইল অনুমতির তিনটে জায়গাতেই ‘r’ দিয়ে রাখতে হবে। রুট, তার গ্রুপ, এবং অন্যরা সকলেই, প্রতিটি ইউজার প্রোগ্রামই যাতে পড়তে পারে। ইউজার প্রোগ্রামগুলো এখান থেকেই সংগ্রহ করে যে তাদের চালাচ্ছে তার সংক্রান্ত খুঁটিনাটি, যখন যখন তাদের ইউজারের পুরো নাম, তার হোম ইত্যাদি তথ্য কাজে লাগছে। একবার ‘`ls -al /dev/passwd`’ মেরে দেখে নিন, আপনারটায় ঠিকঠাক আছে কিনা। হল? দেখেছেন? গুড, এবার ভাবুন, যে কেউ, যে কোনো প্রোগ্রাম পুরোনো /etc/passwd ফাইলটায় এনক্রিপ্টেড চেহারায় পাসওয়ার্ডটাকে দেখতে পাচ্ছে, প্রতিটি লাইনের দ্বিতীয় ফিল্ড থেকে। তার মানে অন্য যে কেউ যে কোনো ইউজারের এনক্রিপ্টেড পাসওয়ার্ড বা ধাঁধানো প্রবেশ-সংকেতটা তুলে নিতে পারছে, এবার সম্ভাব্য প্রতিটা সমাহারকে নিয়ে ট্রাই করে যাচ্ছে। আর এক একটা বড় সিস্টেমে শ-শ ইউজার, তাদের মধ্যে এমন কেবলরাম পাবলিক দু-এক পিস থাকবেই যাদের পাসওয়ার্ড আন্দাজ করা খুব সহজ। নিজের প্রেমিকার জন্মদিন গোছের। অরিজিত বলেছিল দুটো পাসওয়ার্ড শুনে ও খুব মজা পেয়েছিল, একটা সায়মিন্দুর, ও তখন কলকাতা লাগের সার্ভারের ওয়েব-অ্যাডমিন ছিল, পাসওয়ার্ডটা ছিল ‘ami ki jani’। আর আমার সিস্টেমে রুট পাসওয়ার্ড, আগে ছিল, সেটা অবশ্য অতোটা মজার না, ‘shikarer sanket’।

কিছু পাসওয়ার্ড থাকে যারা সরাসরি ডিকশনারি শব্দের সঙ্গে মেলে, এই ধরনের পাসওয়ার্ড ভাঙার জন্যে ডিকশনারি-আক্রমণ হতে পারে, সরাসরি একটা ডিকশনারি ধরে ইংরিজির চালু আশিহাজার শব্দকে পরপর মিলিয়ে যাচ্ছে, এটার চালু নামই হল ‘ডিকশনারি অ্যাটাক’। এ ধরনের প্রোগ্রাম বানানোটাও এমন কিছু শক্ত নয়, যে নিজেই পরপর দেখে যাবে, একের পর এক। কেউ কেউ ডিকশনারি শব্দের আগে পরে একটা করে নম্বর লাগায়, সেটাও খুব সহজেই নিজের অ্যালগরিদমে নিয়ে আসতে পারে পাসওয়ার্ড ক্র্যাক করার সফটওয়ারগুলো। যেটা একজনের মাথায় আসে সেটা আর এক জনের মাথায় আসবেই। যেমন দেখুন স্প্যাম মানে ভাট মেল, এমন চিঠি যা অনাকাঙ্খিত, অপ্রয়োজনীয়, মূলত বিভিন্ন পর্নো সাইটের বিজ্ঞাপন — এই স্প্যাম মেল আটকানোর জন্যে যেসব সফটওয়ার আছে, তারা মেলের সাবজেক্ট লাইনে যে শব্দগুলো আছে, মূলত যৌনশব্দ, সেগুলো থাকলেই সেই মেল আটকে দিচ্ছিল জাংক মেল ফিল্টারে, জঞ্জালচিঠির চিরুনিতে। এই স্প্যামারগুলোও তেমনি, এরা মূলত সফটওয়ার দিয়ে মেল আইডি বানায়, একের পর এক সম্ভাব্য আইডি-তে মেল ছেড়ে যেতে থাকে, আপনি সাড়া দিলেন কী মরলেন, এমনকি পাঠানোতে আপত্তি জানাতেও যদি যান তো, স্প্যামারের কাছে চলে এল যে এই আইডিটা বাস্তব। না-হলেও যে বাঁচা যায় তা নয়। এবার, যেই স্প্যাম ফিল্টারগুলো এই তরকিব কাজে লাগাল, অমনি, স্প্যামাররাও তো বসে নেই, তারা যৌনশব্দগুলোর একটা করে অক্ষর বদলে দিতে লাগলো, উচ্চারণের দিক দিয়ে কাছাকাছি বা আকৃতির দিক দিয়ে কাছাকাছি, ‘o’ বদলে ‘0’, ‘g’ বদলে ‘z’, ইত্যাদি। পাসওয়ার্ড ভাঙিয়ে তথা সিস্টেম ক্র্যাকাররা প্রায়শই বেশ বুদ্ধিমান অল্পবয়স্ক ছেলেপুলে হয়, দুনিয়াকো কুছ কর দিখানার মনোবৈজ্ঞানিক সমস্যায় ভোগে, নিজের যৌন-আত্মবিশ্বাসের অভাবকে বিকিরণ করে দুনিয়াকে তার দিকে তাকাতে বাধ্য করার মধ্যে দিয়ে। অনেক লোকেই এদের ভুল করে হ্যাকার ডাকেন, ‘হ্যাকার’ শব্দটা গ্নু-লিনাক্স তথা ফ্রি সফটওয়ার জগতে একটা সম্মাণের শব্দ। যারা সিস্টেমকে বদলায়, ভাঙে, নতুন আকার দেয়। মূল জায়গাটা এখানে নির্মাণ। আর অন্যের ইউজার অ্যাকাউন্টে অন্যের সিস্টেমে এই বুদ্ধিমান বোকা ছেলেমানুষি নোকঝোঁক — এর নাম ক্র্যাকিং। কখনো কখনো এর মধ্যে বাবা-অবস্থানের প্রতি একটা বিরূপতাও থাকে। ব্যক্তি পারিবারিক বা সামাজিক স্তরে। ভাইরাসও বানায় এরাই, পনেরো থেকে তিরিশ বছর বয়সের, মূলত, একটু সমাজবিমুখ একটা মানসিক রোগে ভোগা পুরুষ ছেলেপুলেরা।

যাইহোক, এই ক্র্যাকারদের হাত থেকে পাসওয়ার্ডগুলোকে বাঁচাতেই মূলত, /etc/shadow ফাইলটা রাখা চালু হয়। এবং আবার মিলিয়ে দেখুন, এই ফাইলটা আর রুট ছাড়া কেউ পড়তে পারেনা, বা রুটের অধিকার সম্পন্ন সিস্টেম ছাড়া, শুধুমাত্র ওই লগ-ইন-এর সময় মিলিয়ে দেখে নেওয়া হয়। মনে আছে, আমরা বলেছিলাম, এই /etc/shadow ফাইলটা কপি করে টেক্সট ফাইল বানাতে আমার কী সমস্যা হয়েছিল? এর বাইরে এই /etc/shadow ফাইলের গঠন প্রকরণ, ফিল্ড সাজানো, যতিচিহ্ন কিন্তু একদম /etc/passwd ফাইলের মতই, পাঁচ নম্বর দিন থেকে মিলিয়ে নিন। এখানে আটটা কোলোন ‘:’ দিয়ে আলাদা করা নটা ফিল্ড বা অবস্থান আছে, তার খুঁটিনাটি অর্থগুলো ‘`man -5 shadow`’ করে দেখে নিন কনফিগারেশন ফাইলগুলোর বিবরণ থাকে ম্যানুয়ালের পাঁচ নম্বর সেকশনে।

৮.৪ | | /etc ডিরেক্টরির আরো দুচারটে ফাইল

/etc/passwd এবং /etc/shadow এই দুটো ফাইলের সঙ্গে আসে আরো দুচারটে ফাইল, এই /etc ডিরেক্টরিতেই, যারা অপারেটিং সিস্টেমের অনুমতি এবং কাঠামোর মালিকানাটা ধরে রাখার কাজে /etc/passwd এবং /etc/shadow ফাইলদুটোকে সহায়তা করে। যেমন, একটা হল /etc/group, এর কাজ ওই গ্রুপের কাঠামোটাকে নিয়ন্ত্রণ করা। বড় বড় সিস্টেমে তো এটা বেজায় কাজে লাগে, ছোট ব্যক্তিগত পিসি সিস্টেমে, যেখানে একাধিক ইউজার আছে, মাঝে মাঝে নিজের রেলাবাজি দেখাতে কাজে আসে, কখনো কখনো সত্যিই নিরাপত্তার জন্যে দরকার পড়ে। এই /etc/group ফাইলে থাকে গ্রুপের বিবরণ। ধরুন আপনার মেশিনে একটা সিডি-বার্নার আছে, এবং আপনি চাননা যে যে কেউ সেটায় সিডি পোড়াক, তখন আপনি 'burner' নামে একটা গ্রুপ তৈরি করতে পারেন, তার কমান্ডটা 'groupadd', ওই 'useradd' কমান্ডের মতই। কমান্ড দিন — '`groupadd burner`'। তবে, মনে রাখুন, ওই 'burner' নামের একটা ইউজার থাকতে হবে, না-থাকলে বানিয়ে নিতে হবে। এবং বিশেষ কিছু ফাইল বা ডিরেক্টরির মালিকানা দিয়ে দিতে পারেন এই গ্রুপকে। যেমন এই সিডি-বার্নারের বেলায় এই ডিভাইস ফাইলটার মালিকানা আপনি দিয়ে দেবেন এই গ্রুপকে, 'chown' কমান্ড দিয়ে। এবং এর পরে, আপনি যাকে যাকে এই গ্রুপে আনতে চান তাদের নাম ঢুকিয়ে দিতে পারেন এই গ্রুপের সঙ্গে ফাইলে। গ্রুপ সংক্রান্ত গোটা খুঁটিনাটি, কী কী গ্রুপ, কোন কোন ইউজার কোন কোন গ্রুপের মেম্বার — এই গোটাটাই লিপিবদ্ধ থাকে /etc/group ফাইলে। আগেই তো বলেছি, একটা গ্নু-লিনাক্স সিস্টেমে সবকিছু, সমস্ত কিছুকে নিজের মত করে বদলে নেওয়া যায় এই ফাইলগুলোকে বদলে। তবে বড় সিস্টেমে এত এত বেশি ইউজার এবং এত এত গ্রুপ থাকতে পারে যাদের এভাবে বদলানো খুব জটিল হয়ে পড়ে। তার জন্যে একটা কমান্ড আছে, 'groupmod', আবার ঠিক 'usermod' কমান্ডের মতই। কোনো একটা ইউজারকে কান ধরে নিজের সিস্টেম থেকে তুলে বাইরে ফেলে দেওয়ার জন্যে যেমন আছে 'userdel', গ্রুপের জন্যে আছে 'groupdel'। এবার ধরুন, আপনি আপনার প্রবন্ধের সংখ্যা এবং পরিমাণে যেমন সৃষ্টিশীল ছিলেন (না না, গুণের কথা বলছি না, প্রবন্ধের গুণগত মান আমি নিজের প্রবন্ধ ছাড়া আর দেখেছি কোথায়? ভাসই তো বোধহয়, যেমন নিজেই লিখেছিলেন, বিপুলা এই ধরিত্রী, কালও নিরবধি, বলা যায়না, ভাসের মত লেখক আবারো তো জন্মাতে পারে কোথাও। এব্যাপারে বস লেখক হলেন নিটশে, তার এক্সে হোমোর একটা চ্যাপ্টারের নাম, হোয়াই আই রাইট সাচ এক্সেলেন্ট বুকস।) ধরুন সেরকমই সৃষ্টিপরায়ন হলেন আপনার গ্রুপ এবং ইউজারদের নিয়ে, গোটাটাই ঘেঁটে ফেলেছেন, এবার আপনার সেই গ্রুপিণ্ডটা পাবেন কোথায়? ওই /etc/group ফাইলটা তো আছেই, কিন্তু বড় ফাইল হলে তার মধ্যে খুঁজে খুঁজে বার করা কোন কোন ইউজার কোন কোন গ্রুপের মেম্বার — সে প্রায় অসম্ভব। তার জন্যে একটা কমান্ড আছে 'groups'। . . . *খেজুরগাছে হাঁড়ি বাঁধো ম্যান . . .*

/etc ডিরেক্টরিতে শুধু /etc/passwd, /etc/shadow, আর /etc/group এই কনফিগারেশন ফাইলগুলোই নয় একটা আস্ত ডিরেক্টরি থাকে এই ইউজারের খুঁটিনাটির কাজে, তার নাম '/etc/skel'। এই স্কেলিটন বা কংকালটা একটা টেমপ্লেট, একটা ছক, একটা ছাঁচ। এই টেমপ্লেট ধারণাটা গোটা কম্পিউটার মননেই বারবার আসে, অনেকটা সময় ধরে ভাবলাম একটা জবর প্রতিতুলনার জন্যে, তেমন কিছু মাথায় এলনা। সবচেয়ে কাছাকাছি আসছে যেটা সেটা এরকম। ধরুন একটা মেয়ে, তার একটা টেমপ্লেটের নাম কনে। তখন তার পরনে লাল বেনারসি, কপালে চন্দন, গায়ে গয়না, হাতে কাজললতা, ইত্যাদি। এবং এই টেমপ্লেটটা মুহূর্তিক বা ইনস্ট্যানটেনিয়াসও নয়। কারণ কনে সাজার একটা দীর্ঘ শ্রমসাধ্য প্রক্রিয়া আছে, গায়ে হলুদ, চুলে হেনা, ঠোঁটে তাম্বুল-না-হলেও-রেভলন, ইত্যাদি। মানে এই টেমপ্লেটে ঢুকবে একটা মেয়ে, বেরোবে কনে হয়ে। আবার ওই মেয়েকে যখন রাঁধুনি টেমপ্লেটে দেখছি, তখন তার পরনে ছাপাশাড়ি গাছকোমর করা, চুল উড়োঝুরো, গালে একটু ময়দা, নাকে ঘাম, আঙুলে হলুদ, হাতে খুন্তি, ইত্যাদি। এখানেও একটা প্রক্রিয়া আছে, সে এসেছে তরকারি কোটা, মাছে হলুদ-নুন মাখানো, ময়দা মাখা ইত্যাদি উপাদানে তৈরি একটা টেমপ্লেটের ভিতর দিয়ে — ওই একই মেয়ে এখন রাঁধুনি। /etc/skel ডিরেক্টরিতে থাকে ঠিক এইরকম একটা টেমপ্লেট, একজন ইউজারের হোম ডিরেক্টরির। যেখানের প্রতিটি কনফিগারেশন ফাইলকে কপি করে দেওয়া একজন নতুন ইউজার যোগ করার সময় তার ইউজার ডিরেক্টরিতে। এই ফাইলগুলোর মধ্যেই দেওয়া থাকে বিভিন্ন সময়ে সিস্টেমের সঙ্গে তার সম্পর্ক কী হবে, তার লগ-ইনের পরপরই সিস্টেম কী করবে, তার কমান্ড প্রম্পট কী হবে, তার কিছু জরুরি ইউজার প্রোগ্রাম চালানোর সময়ে সিস্টেম কোন কোন ডিরেক্টরির কোন কোন ফাইল খুলবে,

তার এক্স উইনডোজ চালানোর খুঁটিনাটি, ওই ইউজারের এনভায়র্নমেন্ট ভ্যারিয়েবলগুলো কী হবে, ইত্যাদি। এই এনভায়র্নমেন্ট ভ্যারিয়েবলগুলোর কথায় আমরা পরে আসব।

৯।। লিংক

ছ নম্বর দিনের সেকশন ৭-এর ডিভাইস ফাইলগুলোকে আমাদের বুঝতে অসুবিধে হচ্ছিল বলে আমরা গিয়েছিলাম ফাইলের মালিকানা এবং অধিকারের আলোচনায়। সেটার প্রাথমিক একটা ধারণা আমাদের হয়েছে, এখনো ওই ডিভাইস ফাইলগুলোকে বোঝার একটা সমস্যা রয়ে গেছে, তার নাম লিংক, দেখুন যে ফাইলগুলোর লং বা দীর্ঘ লিস্টিং শুরু হয়েছে '`l`' দিয়ে, যেমন ব্লক ডিভাইসের '`b`', ক্যারেকটার ডিভাইসের '`c`', ডিরেক্টরির '`d`' এবং রেগুলার বা সাধারণ ফাইলের বেলায় নিছক '`-`'। এই লিংক ব্যাপারটাকে একটু বুঝে নেওয়া যাক। প্রায়ই একটা ফাইলের একই সঙ্গে দুজায়গায় বা দুইয়ের বেশি জায়গায় থাকার দরকার পড়ে। নানা ধরনের প্রোগ্রাম নানা কাজে নানা জায়গায় তাদের ফাইল খোঁজে। এবার কোনো একটা বিশেষ ফাইল যদি এরকম একাধিক প্রোগ্রামের সঙ্গে অন্বিত হয় তাহলে তার একই সঙ্গে অনেক উপস্থিতির দরকার পড়ে। এর মধ্যে অনেকসবয় কিছু ঐতিহাসিক কারণও থাকে, আগে একসময় প্রথা ছিল একটা বিশেষ ফাইলের একটা বিশেষ জায়গায় থাকা, পরে কাজের সুবিধের জন্যে জায়গাটা বদলানো হয়েছে, এবার পুরোনো সময়ের প্রথায় তৈরি প্রোগ্রামগুলো সেই ফাইলটাকে পাবে কী করে? একই সঙ্গে কেউ একের বেশি জায়গায় থাকতে পারেনা। তাই, একটা জায়গায় সত্যিই নিজেকে রেখে, অন্য জায়গাটায় বা জায়গাগুলোয় নিজের বদলে নিজের একটা কান রেখে আসার দরকার পড়ে, পরে কান টানলেই যাতে মাথা আসতে পারে। এইসব কানবাজি না করে সরাসরি আর এক জায়গায় ফাইলটা কপি করে দেওয়ার সমস্যা এই যে কপি করা মাত্রই কপিটা তো নিজেই একটা স্বতন্ত্র ফাইল হয়ে গেল, নিজের অস্তিত্ব এবং ইতিহাস। তার মানে সিস্টেমের হিশেব রাখতে হবে দুটোর মিলের, একটা বদলালেই অন্যটায় তার অনুরূপ বদল ঘটাতে হবে। সিস্টেমের উপর আর কত অত্যাচার করা হবে? আর সিস্টেম যে চালায়, মানে সেই অ্যাডমিন্সট্রেটর বা সুপারভাইজর বা সুপারইউজার, তাদের অবস্থাটা ভাবুন। এবার একটা ফাইলকে ভাবুন যা একই সঙ্গে অনেক ডিরেক্টরিতে আছে, কিন্তু একটা জায়গায় তার ভিতরকার তথ্য বদলালেই অন্য জায়গাগুলোয় বদলে যায়। মানে এক কথায় এটা যেন একাধিক ফাইল, যাদের দুজনেরই তথ্যের ভাণ্ডারটা এক। এই কাজটাই করে লিংক, এই রকম একই তথ্যের একাধিক ফাইল বানানোর সুযোগ দেয়।

কাজটা করে দেখা যাক। আমরা একাধিকবার 'cat' বা কনক্যাটেনেট, মানে যোগ করা বা সাজানো, কমান্ডটাকে ব্যবহার করেছি কিছুটা পরিমাণ টেক্সটকে পেতে দেওয়ার কাজে। কখনো পেতে দিয়েছে স্ক্রিনে, আমরা কনসোলে ফাইলটা পড়েছি। যেমন '`cat lilo.conf`' বলে আমরা 'lilo.conf' ফাইলটাকে স্ক্রিনে পেতে নিয়ে পড়েছি। বা '`cat lilo.conf>lilo.text`' বলে আমরা সেই 'lilo.conf' ফাইলের টেক্স্টটাকে পেতে দিয়েছি 'lilo.text' ফাইলে। এবার ফাইল থেকে স্ক্রিনে পেতে দেওয়ার ঠিক উল্টো প্রক্রিয়ায় স্ক্রিনে লেখা আমাদের টেক্সটকে ফাইলে পেতে দিতে দেখব আমরা 'cat' কমান্ডটাকে। ছোটখাটো দু-চার লাইনের ফাইল লিখবার জন্যে এটা একটা ভালো উপায়।

'`cat > onefile`' কমান্ড লিখে এন্টার মারার পর, কমান্ড প্রম্পট আর ফেরত আসবে না। কমান্ড প্রম্পট ফেরত না-আসার মানে কী সেটা মনে করুন পাঁচ নম্বর দিন থেকে। তার মানে সিস্টেম আপনার কাজে নিয়োজিত, কাজটা চলছে, এখনো সমাপ্ত হয়নি, সমাপ্ত হলেই কমান্ড প্রম্পটটা ফেরত আসত। এবার ঠিক টাইপ করার লাইনের নিচেই কালো স্ক্রিনে আপনি যা টাইপ করবেন সেটা ওই 'onefile' ফাইলে যোগ হয়ে চলেছে। আপনি একটা লাইন লিখে এন্টার মারলে সেই নিউলাইন-চিহ্নটাও ঢুকে যাচ্ছে ওই ফাইলে, পরে সে ওই ভাবে লাইন ভেঙেই দেখাবে। এই টাইপ করা টেক্সটকে ফাইলে তোলার প্রক্রিয়াটা চলতেই থাকবে যতক্ষণ না আপনি 'কন্ট্রোল-ডি', <Ctrl-D>, মারছেন, মানে 'কন্ট্রোল' সুইচটা টিপে রেখে 'ডি' সুইচটা টিপছেন। এবার 'কন্ট্রোল-ডি' মেরে বেরিয়ে এসে আপনি '`ls`' মারলেই আপনি দেখতে পাবেন 'onefile' ফাইলটাকে। এবার কমান্ড দিন '`cat onefile`', দেখুন স্ক্রিনে ফুটে উঠেছে হুবহু যা আপনি টাইপ করেছিলেন। প্রথম 'cat' আপনার টাইপটাকে রিডাইরেক্ট করে পৌছে দিল স্ক্রিন থেকে 'onefile' ফাইলে, দ্বিতীয়বার 'onefile' ফাইল থেকে স্ক্রিনে। এই 'cat' কমান্ড দিয়েই আবার টেক্সটের সঙ্গে নতুন টেক্সট যোগও করা যায়। আগের বারের '`cat > onefile`' কমান্ডটার জায়গায় এবার নতুন কমান্ড দিন

"cat >> onefile"। আবার টাইপ করে যান। একই ভাবে 'কন্ট্রোল-ডি' মেরে বেরিয়ে আসুন। আবার করুন 'cat onefile', এবার দেখুন, আগের টেক্সট-এর সঙ্গে পরেরটুকু যোগ হয়ে গেছে। এর নাম অ্যাপেন্ড করা। এর চিহ্ন হল '>>', যেরকম চালান করার বা রিডাইরেক্ট করার চিহ্ন হল '>'। শুধু স্ক্রিন থেকে রিডাইরেক্ট করে অ্যাপেন্ড করা নয়, সরাসরি একটা ফাইলকেও অ্যাপেন্ড করা যায়। এইমাত্র বলা প্রক্রিয়ায় আর 'cat > twofile' কমান্ড লিখে একটা ফাইল লিখুন 'twofile'। ফাইল বানানো শেষ হওয়ার পরে, কমান্ড দিন 'cat twofile >> onefile'। তারপর গোটাটাকে স্ক্রিনে ফুটিয়ে তোলার জন্যে কমান্ড দিন 'cat onefile'। দেখুন, 'twofile' ফাইলের টেক্সট যোগ হয়ে গেছে 'onefile' ফাইলের মধ্যেই, পুরোনো টেক্সট-এর পরে। 'touch' দিয়ে আমরা যেমন ফাঁপা ফাইল বানানোর কায়দা শিখেছিলাম, তেমনি গ্নু-লিনাক্সে টেক্সট ফাইল বানানোর সহজতম কায়দাটা আমরা এইমাত্র শিখলাম। এবার এটাকেই আমরা ব্যবহার করব লিংক বোঝার কাজে।

৯.১।। সিম্বলিক বা সফট লিংক

প্রথমে টাচ করুন 'onefile' নামের একটা ফাঁপা ফাইলকে। কমান্ড দিন 'touch onefile'। এবার কিছু আলাদা করে না বুঝেই পরের কমান্ডটা দিন, পরে আমরা এর মানেটা বুঝব — 'ln -s onefile twofile'। কমান্ড প্রম্পট তো ফেরত এলো তার কাজ শেষ করে, কিন্তু এই কমান্ডে কাজটা কী হল সেটা বুঝতে হবে তো, ডিরেক্টরির হালহদিশ বোঝার জন্যে আমাদের কমান্ড তো আছেই, 'ls -al'। স্ক্রিনে দেখুন, পরিষ্কার দুটো ফাইলকে দেখাচ্ছে। তার মানে আমাদের ওই না বোঝা কমান্ডটা আমাদের টাচ করা 'onefile' ফাইলটা ছাড়াও একটা নতুন ফাইল তৈরি করেছে 'twofile' নামে, এবং এদের দুজনের বাইট সাইজটা দেখুন তো? চমকালেন? এর মানে কী? 'onefile' দেখাচ্ছে শূন্য, অথচ 'twofile' দেখাচ্ছে সাত? 'onefile' শূন্য এর মানেটা তো বুঝতে পারছি, আমরা কিছুই রাখিনি সেখানে, শূন্য ফাঁপা শুধু নামসর্বস্ব, যাকে জরুরি অবস্থার পরের ফেজে আমরা ডাকতাম সিদ্ধার্থ, চায়ের দোকানে গিয়ে বলতাম, মেনোদা দুটো চা, একটা সিদ্ধার্থ, মানে দুটোকে তিনটে করো। মধ্যে এতগুলো বছরে কিশোর সিদ্ধার্থ বড় হয়েছে, এখন বোধহয় বলা যায়, দুটো চা, একটা বুদ্ধদেব। কিন্তু ফাইল সাত দেখাচ্ছে কেন? ফিরে আসব এটায়, মাথায় রাখুন।

এবার, আমরা জানি 'cat' দিয়ে কী করে ফাইলে টেক্সট ভরতে হয়, ভরুন আপনার মনোমত টেক্সট, যা খুশি, কিন্তু সাবধান, গ্নু-লিনাক্স বিষয়ে কোনো খারাপ কথা লিখবেন না, এটা গ্নু-লিনাক্স সিস্টেমের একটা বিশেষ বৈশিষ্ট্য যে আপনি যদি প্রকৃত বিশ্বাস থেকে কোনো খারাপ কথা সিস্টেমের বিষয়ে লেখেন আপনার স্ক্রিনে, সঙ্গে সঙ্গে আপনার মনিটরটা সাউন্ডবক্স হয়ে যাবে আর মড়াকান্না কাঁদতে থাকবে, হার্ডডিস্ক পুরো ফরম্যাট করে ফেলার আগে অব্দি। টেক্সট ভরার পর 'কন্ট্রোল-ডি' মেরে বেরিয়ে আসুন। আমরা জানি, এখন স্ক্রিনে 'cat onefile' কমান্ড দিয়ে আমরা এইমাত্র টাইপ করা গোটা টেক্সটটা দেখতে পাব। কিন্তু, একই কাজ কবার করতে ভালো লাগে মানুষের? কমান্ড দিন, 'cat twofile', দেখুন তো কী ঘটেছে? স্ক্রিনে আপনি দেখছেন হুবহু 'onefile' ফাইলে ঢোকানো টেক্সটটা। হ্যাঁ, ঠিক এটাই ঘটার কথা ছিল। দুটো আলাদা ফাইল, তাতে তথ্য একই। ঠিক এটাই করে আমাদের ওই নতুন কমান্ড 'ln -s'। আসলে 'ln' হল কমান্ড, আর '-s' তার অপশান। 'ln' কমান্ডটা একটা ফাইলের লিংক তৈরি করে, মানে, অন্য এমন একটা ফাইল তৈরি করে যা প্রথম ফাইলের তথ্যটাই বহন করে, তাতে কোনো বদল এলে লিংক ফাইলটাও বদলে যায়। এবং একবার 'ls -al' মেরে দেখুন তো, দীর্ঘ তালিকায়, আমরা যেমন দেখলাম, একদম বাঁদিকের জায়গাটায়, এক একটা অক্ষরের এক একটা মানে, এই দুটো ফাইলের, 'onefile' আর 'twofile', অক্ষরদুটো আলাদা কী কী? কী দেখছেন? যখন 'onefile', তখন '–', আর যখন 'twofile', তখন 'l'। এর মানেটা কী — মনে করুন। তিন প্যারা আগেই আছে। মানে, 'twofile' হল একটা লিংক, সিম্বলিক লিংক, যার নিজের শরীরে আসলে কিছুই নেই, শুধু একটা দিকচিহ্ন, দেখিয়ে দেওয়া সেই ফাইলটাকে যার সে লিংক। যখন আমরা এই সিম্বলিক লিংক ফাইলকে ক্যাট করছি, ডগ করছি, যা-খুশি করছি, তখন আসলে তাকে কিছুই করছি-না, তার শরীরে পাওয়া লিংক বেয়ে সিস্টেম পৌছে যাচ্ছে যার লিংক সে সেই আসল ফাইলে। যা যা আমরা ঘটাচ্ছি, সবই ঘটছে ওই মূল ফাইলে। লিংক ফাইলের নিজের বলে কিছু নেই শুধু ওই মূল ফাইলটাকে দেখিয়ে দেওয়া ছাড়া, তোমারি গরবে গরবিনী আমি, রূপসী তোমারি রূপে।

'`ls -al`' কমান্ডের কাজ আমাদের এখনো ফুরিয়ে যায়নি, এবার এই দুটোর বাইটসাইজ দেখুন তো? 'onefile' কত তা আমি বলতে পারব না, যতটা আপনি লিখেছেন। কিন্তু 'twofile' কত তা আমি নিশ্চিতভাবেই জানি — সাত। কী করে? এবার আর একটা ফাঁপা ফাইল বানান, আপনার কাছে কোনো ব্যাপার না, আপনি তো টাচউড হয়ে গেছেন, ছুঁলেই ফাইল। শুধু এবার তার নাম দিন 'threefile', এবার তাকে '`ln -s`' করে আর একটা ফাইল বানান, তার নাম দিন 'fourfile'। আবার যদি '`ls -al`' করেন, এবার 'threefile' যথারীতি পাচ্ছেন শূন্য, 'fourfile' কিন্তু বাইটসাইজ পাচ্ছেন নয়, আগেরবারের মত সাত না। এবার 'threefile' ফাইলে টেক্সট ভরুন, যত ভরবেন সেই অনুযায়ী তার বাইটসাইজ বাড়বে, কিন্তু 'fourfile' সেই দশের থেকে এক কম থেকে যাবে চিরকাল। এই সাইজটা আমি বলছি কী করে? বুঝতে পারছেন? না-পারলে শূন্য নম্বর দিনের একদম শেষ দিকের টেক্সট-তথ্যের সাইজের জায়গাটা আর একবার পড়ে আসুন, তাতেও যদি আপনার মাথায় না-এসে তার মানে এত সময় ধরে বোকার মত পড়ে পড়ে আপনার বেসাল মেটাবলিক রেট কমে গেছে। তারস্বরে, সাবধান যদি রাত্তির হয়, হেডফোন লাগিয়ে, একটু সুখবিন্দার সিং বা জসপিন্দার নরুলা শুনুন। আচ্ছা, আপনাকে একটু হেল্প করে যাই, দেখুন তাতে পারেন কিনা — এবারে আরো দুটো ফাইল বানান, ওই একই ভাবে, প্রথমে টাচ, তারপর লিংক। এবারে নাম দিন 'fivefile' আর 'sixfile'। এবারেও, 'fivefile' অবভিয়াসলি পাবেন শূন্য, কিন্তু 'sixfile' পাবেন আট।

শুধু ফাইলই না কিন্তু, এই সিম্বলিক লিংক তৈরি করা যায় একটা ডিরেক্টরিরও। তখন ওই সিম্বলিক বা প্রতীকী লিংকের উপর আমাদের যে কোনো কাজ আসলে ঘটবে সেই ডিরেক্টরির উপর, যার সে লিংক। ধরুন আপনার ডিরেক্টরি 'onedir', এর একটা সিম্বলিক বানালেন 'twodir'। এবার এই 'onedir', মূল ডিরেক্টরির মধ্যে একটা ফাইল বানান, আপনি বানাতে জানেন, ফাঁপা ফাইল মানে 'touch', আর দোকানে কোনো মাল রাখতে চাইলে 'cat'। এবার '`ls -al twodir`' করে দেখুন আপনার এইমাত্র বানানো ফাইলের খুঁটিনাটি অবিকল দেখিয়ে দিচ্ছে, যা আপনি বানিয়েছিলেন 'onedir' ডিরেক্টরিতে। আমাদের কাছে 'twodir' আসছে মূল 'onedir' ডিরেক্টরির একটা হুবহু প্রতিলিপি হয়ে, কিন্তু এটা কোনো প্রতিলিপি না, আমরা আসলে 'onedir' ডিরেক্টরিকেই দেখছি, শুধু অন্য নামে।

এই নানা নামে ডাকা গোলাপের গল্প নিছক ফুল ভালোবাসা নয়, এর নানা জরুরি ব্যবহার আছে। দু-একটা আমরা দেখব। অন্যগুলো দেখতে থাকবেন, সিস্টেমে কাজ করতে করতে। যেমন ধরুন, আপনার দুটো সিডি-ড্রাইভ আছে, একটা বার্নার, আর একটা নিছক রিডার। আপনি বার্নারটাকে শুধু সিডি পোড়ানোর কাজে ব্যবহার করতে চান, আর এমনি রিডারটাতেই ভিউসিডি অডিয়োসিডি ইত্যাদি চালাতে চান। দুটো সিডিড্রাইভ, সেটা যদি স্কাসি হয় তাহলে, ছ নম্বর দিনে দেখেছি আমরা, তার নাম '/dev/scd0' আর '/dev/scd1'। আবার যদি তারা আইডিই ড্রাইভ হয়, তাকে পাব '/dev/sr0' আর '/dev/sr1' ফাইলে। সচরাচর এর প্রথমটাই হয় বার্নার ড্রাইভ, কারণ, মাদারবোর্ডের মাস্টার সংযোগে বার্নার আর স্লেভ সংযোগে রিডার না-লাগালে কিছু অসুবিধা হতে পারে, তার মানে '/dev/scd1' বা '/dev/sr1' হল সেই ড্রাইভ যেখান থেকে আপনি ভিউসিডি বা অডিয়োসিডি চালাতে চান। এবার ধরুন, আপনি এমপ্লেয়ার বা যেকোনো ইউজার প্রোগ্রাম চালাচ্ছেন, যার ডিফল্ট সেটিং করা আছে — তুমি সিনেমার ভিউসিডি খোঁজো '/dev/cdrom' ফাইলে, সেখানে যা পাবে তাকে চালাও। কোনো অসুবিধে নেই, আপনি একটা সিম্বলিক লিংক বানিয়ে দিন, কমান্ড দিন 'ln -s /dev/scd1 /dev/cdrom' বা, 'ln -s /dev/sr1 /dev/cdrom', মানে আপনারটায় যা, সেই অনুযায়ী। প্রথম প্রথম যখন ডিভাইস ফাইল দিয়ে এটা বুঝতে অসুবিধে হয়, বা সিস্টেমের সঙ্গে আপনার পরিচিতি যথেষ্ট নয়, তখন একটা সরল উপায় হল 'eject /dev/scd1' বা 'eject /dev/sr1' মারা — একটা একটা করে ডিভাইস ফাইল দিয়ে দেখুন, কোনটায় আপনার আজি সিডিরো দুয়ার খোলা হচ্ছে, সেটাই আপনার ডিভাইস ফাইল। ব্যাস, যেই সিম্বলিক লিংকটা তৈরি হল, কাজ খতম, এখন থেকে সিস্টেম নিজেই যখন '/dev/cdrom' খুঁজবে, আসলে পাবে '/dev/scd1' বা '/dev/sr1' ডিভাইস ফাইলকে। এবার ছ নম্বর দিনের তুলে দেওয়া পনেরোটা ডিভাইস ফাইলের দীর্ঘ তালিকা বা লং লিস্টিং থেকে দেখুন তো এই কায়দাটা আপনি বুঝতে পারছেন কিনা, তখন যেটা আমরা ছেড়ে এসেছিলাম? লিংক ফাইলের মানে বুঝতে পারছেন? পরে দেখবেন, যখন আমরা মাউন্ট ব্যাপারটা ভালো করে বুঝব, মাউন্টের ব্যাপারেও খুব কাজে লাগে সিম্বলিক লিংক। এবং সিস্টেমে গোড়া থেকেই কোথায় কোথায় কোন ফাইলের বা ডিরেক্টরির সিম্বলিক লিংক থাকবে তার ছকটা করা থাকে গ্নু-লিনাক্সে ফাইলসিস্টেম হায়েরার্কি স্ট্যান্ডার্ডের মধ্যেই।

৯.২।। সফট লিংক আর হার্ড লিংক

গ্নু-লিনাক্সে একই ফাইল একটা ফাইলসিস্টেমের একাধিক জায়গায় একাধিক নামে থাকতে পারে, যেখানে তথ্যের ভাণ্ডারটা একই শুধু নানা নাম তাদের, একে বলে হার্ড লিংক। সিম্বলিক লিংকের সঙ্গে এর মিল আছে, কিন্তু পার্থক্যও আছে। দুই রকম লিংক-কে একসঙ্গে পেড়ে ফেলা যাক। শুধু, সরি, এটার জন্যে একটু হুবহু টাইপ করতে হবে, যারা টাইপ করতে খুব একটা অভ্যস্ত না, তাদের টাইপ করার সময় একটা কেলো হয় এই যে, বর্ণমালাগুলো ঠিকঠাক আসে, কিন্তু যতিচিহ্ন আর স্পেসগুলো ঠিকঠাক খেয়াল থাকেনা। এখানে একটা বাড়তি যতিচিহ্নও খেয়াল রাখবেন, নিউলাইন মানে নতুন-লাইন, সচরাচর টেক্সট-এর ভিতর আমরা এন্টার বা ক্যারেজ-রিটার্ন মেরে যা করি — যেখানে লাইন শেষ করে এটা মারতে হবে সেখানে আমি বলে দেব।

প্রথমে টাচ করে ‘onefile’ বানান, ‘`touch onefile`’। এবার দুরকমের দুটো লিংক বানান, প্রথমটা সিম্বলিক বা সফট লিংক ‘twofile’, কমান্ডটা তো ইতিমধ্যেই জানেন, ‘`ln -s onefile twofile`’। এবার দ্বিতীয় একটা লিংক বানান, ‘threefile’, এটা কিন্তু হার্ড লিংক, এবারে কমান্ড দিন ‘`ln onefile threefile`’। এবার একবার ‘`ls -ail`’ মারুন, খেয়াল করুন, এই অপশানটা ‘`-ail`’ কিন্তু আমাদের অভ্যস্ত অপশান ‘`-al`’ থেকে আলাদা, মধ্যের বাড়তি ‘`i`’ অপশানটা কী দেখাচ্ছে সেটা আমরা এখুনি দেখতে পাব। যে তালিকাটা ফুটে উঠবে সেটা এইরকম, শুধু বোঝার সুবিধের জন্যে পরের দুটো লাইন আগে পরে করে দিয়েছি, নামের বর্ণমালা অনুযায়ী এলএস তো থ্রিফাইলকে টুফাইলের আগে এনেছিল।

```
12623010 -rw-r--r-- 2 dd users 0 2003-12-24 10:49 onefile
12623012 lrwxrwxrwx 1 dd users 7 2003-12-24 10:50 twofile -> onefile
12623010 -rw-r--r-- 2 dd users 0 2003-12-24 10:49 threefile
```

এরমধ্যে একদম বাঁদিকের ওই বাড়তি গাবদা নম্বরটা ওই ‘`i`’ অপশানের অবদান। আইনোড নম্বর। আসছি আমরা সেই কথায়। এখানে দেখুন, সফট লিংক বা ‘twofile’-এর গল্পটা আমাদের চেনা, ঠিক এরকমই এসেছিল আগে, বাইটসাইজ সাত, আর বাঁদিকে লেখা ‘`l`’ কিন্তু, হার্ড লিংক বা ‘threefile’ দেখুন, কিছু মেলাতে পারছেন? সবকিছুই ‘onefile’-এর মত, যেন নিজেই একটা স্বতন্ত্র স্বাধীন ফাইল, শুধু একটা জায়গা বদলে গেছে দেখুন, আজকের সেকশন ৮.১ থেকে মিলিয়ে নিন, লিংক-এর সংখ্যাটার জায়গাটা বদলে গেছে, এখন ‘onefile’ আর ‘threefile’ দুটোর বেলাতেই এই সংখ্যাটা এসেছে দুই। আরো একটা জিনিষ দেখুন সমস্ত দিক থেকে সমস্ত অর্থে ‘onefile’ আর ‘threefile’ হুবহু এক। কী মজা দেখুন, আমরা তো প্রথমে বানালাম ‘onefile’, তারপরে ‘twofile’, সবার শেষে ‘threefile’। সময়ের স্ট্যাম্পটা দেখুন, ‘twofile’-এর বেলায় ‘`10:50`’, ঠিকই আছে, ‘onefile’-এর ‘`10:49`’ থেকে একমিনিট পরে। কিন্তু এমনই গোলমেলে ওই ‘threefile’ যে সে কিন্তু তার নিজের বাপের মানে ‘onefile’-এর সঙ্গে একই ডেট-অফ-বার্থ টাইম-অফ-বার্থ শেয়ার করছে, কেলেংকারির একশেষ। আসলে বাপ-ব্যাটার সম্পর্ক নয়, সম্পর্কটা একদম হুবহু প্রতিরূপের। এবার আসুন একদম বাঁদিকের আইনোড নম্বরে, সেটা কী বস্তু মনে আছে? আগের ছ নম্বর দিনে একটু আলোচনা আছে, আরো ভালো করে আজই আসবে। আমরা যাকে নামে চিনি, সেই ফাইলটাকে অপারেটিং সিস্টেম যা দিয়ে চেনে তাই হল আইনোড নম্বর। শুধু এইটা দেখুন, ‘onefile’ আর ‘threefile’-এর আইনোড নম্বরটাও এক, যা কিন্তু ‘twofile’-এর বেলায় সত্যি নয়। ঠিক একই জিনিষ দেখুন অনুমতির কাঠামোয়। ‘onefile’ আর ‘threefile’ এক মানে এক মানে এক।

এবার দেখা যাক ‘onefile’-এ কিছু তথ্য যোগ করে। প্রথমে টাইপ করুন, আগেও করেছেন, ‘`cat > onefile`’, এন্টার মারুন। দেখুন কমান্ড প্রম্পট হাওয়া হয়ে গেছে, নিচের লাইনে কার্সরের চৌকো আলোটা দপদপ করে আপনাকে ডাকছে, এসো এসো টাইপ করো। সত্যিকারের লিনাক্সী হয়ে ওঠার প্রমাণ হল যখন আপনি ডাকটা কানেও শুনতে পাবেন, তার জন্যে দরকার রোজ জিটাইপিস্ট (Gtypist) নিয়ে প্রাকটিস করা। গ্নু-র একটা ফ্রি সফটওয়ার। টাইপ শেখার ইশকুলে যা শেখায় তার চেয়ে তেরো হাজার পাঁচশো বারো গুণ ভালো করে শেখায়। আর বকে দেওয়াটা যা কিউট, সে আর কহতব্য নয়। ওই চৌকো আলোর জায়গাটায় আপনি হুবহু টাইপ করুন, স্পেস মানে স্পেস, অক্ষর মানে অক্ষর, ‘`onefile is one file`’। এবার এন্টার মারুন। মানে নিউলাইন, নতুন লাইনে চলে গেছে আমন্ত্রণী কার্সর। এবার আপনি জানেন কী করে ফাইলটা গোটাতে হবে, ‘কন্ট্রোল-ডি’ মেরে, মানে ‘Ctrl’ সুইচটা টিপে রেখে ‘D’ সুইচটা মেরে, তারপর দুটোই ছেড়ে দিয়ে। এবার একবার ‘`cat onefile`’ করে দেখে

নিন, হুবহু এই টেক্সটটাই দেখাচ্ছে স্ক্রিনে। আপনি আপন টাইপমাধুরী মিশায়ে অন্য কিছু ভরতেই পারতেন ফাইলটায়, কিন্তু তাহলে টেক্সট-এর চিহ্নের হিশেবটা আবার বদলে যেত, হিশেবটা এখুনি আমাদের দরকার পড়বে। তার আগে একটু কুচো করে টাইপিং শিখে আসা যাক, ব্যাশ কী ভাবে আপনার ইতিহাসটাকে লিখে রাখে আপনারই অগোচরে, আপনার ইতিহাস মানে আপনার আদেশের ইতিহাস। সেই ইতিহাসকে কী করে প্রয়োজন মাফিক ফেরত পাওয়া যায়।

৯.৩।। আদেশের ইতিহাস

আপনার কিবোর্ডের তিনখাবলা চাবিসংস্থানের মধ্যে একদম ডানদিকেরটা তো নিউমেরিক কিপ্যাড, অব্যবহারের ধুলো জমে জমে এমন কালো হয়ে যায় যে মাসআটেক বাদে বাদে যখন সুইচগুলো খুলে সাবানজলে ভেজাই, ওগুলোকে আলাদা করে স্কচব্রাউট দিয়ে মাজতে হয়। একদম বাঁদিকের মূল অংশটা হল সেই জায়গাটা আমরা প্রায় সব টাইপই যেখানে করি, উপরে সংখ্যার সারি, নিচে বর্ণমালা আর নানা চিহ্ন। আর এই দুইয়ের মাঝে এক খাবলা সুইচ। উপরে ছটা, নিচে চারটে। নিচের চারটে হল অ্যারো সুইচ।

এবার আপনাকে অসম্ভব শক্ত জটিল দুঃসাধ্য দুঃসাহসী রোমহর্ষক একটা টাইপিং করতে হবে, এই চারটে অ্যারো সুইচের মধ্যে চাতক সুইচটিকে, মানে যে অ্যারোটা উপরপানে মুখ করে আছে, ক্রিচ করে টিপতে হবে একবার করে, থেমে স্ক্রিন দেখতে হবে। একবার ক্রিচ করে থাকলে — এখন স্ক্রিনে আছে '`cat onefile`'। ক্রিচ ক্রিচ করে থাকলে স্ক্রিনে আছে '`cat > onefile`'। ক্রিচ ক্রিচ ক্রিচে '`ls -ail`'। আপনি আরো ক্রিচী মানুষ হতে চাইলে পরপর পেতে থাকবেন, পরপর, '`ln onefile threefile`', তারপর আসবে '`ln -s onefile twofile`', তারপর '`touch onefile`'। এইভাবে চলতেই থাকবে। আজ যা করেছেন, তার আগে যা করেছেন, তার আগে, তার আগে, গতজন্মেও চলে যেতে পারে, যদি আপনি তখন থেকে গ্নু-লিনাক্স শুরু করে থাকেন। এবং গতজন্ম থেকে এই অব্দি এক হাজারের বেশি কমান্ড না-দিয়ে থাকেন। কারন কমান্ড মনে রাখার এই প্যাঁচটা হল এক হাজারি মনসবদার। এর জমিদারি আরো বাড়ানো যায়, কী করে সেটা পরে শিখবেন।

এই দুঃসাধ্য ক্রিচকৌশলের আরো একটা উপায় আছে, 'কন্ট্রোল-আর', মানে 'Ctrl' সুইচটা টিপে রেখে 'R' সুইচটা মেরে, তারপর দুটোই ছেড়ে দেওয়া। এবার যে কমান্ডটা আপনি খুঁজছেন, ধরুন '`ls`', সেটাকে টাইপ করে দিন, '`l`' মারা মাত্র শেষ যে কমান্ডে আপনি '`l`' দিয়ে শুরু করেছেন সেটা চলে আসবে স্ক্রিনে, তারপর '`s`' মারা মাত্র শেষ যে কমান্ডে আপনি '`ls`' দিয়ে শুরু করেছেন সেটা চলে এসেছে। এবার আপনি ধরুন দেখলেন যে এটা নয়, এরও আগে আপনি যে '`ls`' ব্যবহার করেছেন সেটা পেতে চান, কোই বাত নেহি, আর একবার 'কন্ট্রোল-আর' (Ctrl-R)মারুন, তার আগেরটায় যেতে চাইলে আরো একবার, এই করতে করতে যেই ঠিক কমান্ডটায় পৌঁছলেন তখন আর একবার ক্রিচ করে দক্ষিণপন্থী অ্যারো সুইচটা টিপুন। দেখলেন ওই কমান্ডটা চলে এসেছে স্বাভাবিক কমান্ড প্রম্পটে, এখন এন্টার মারলেই কমান্ডটা কাজ করবে, যেন এইমাত্র আপনি টাইপ করে দিয়েছেন। এবার ধরুন দেখলেন যে কমান্ডটা মোটামুটি ঠিক আছে, একটু বদলাতে হবে। বামপন্থী বা দক্ষিণপন্থী তীরেরা আছে কি করতে। ওই দিয়ে কার্সর এগিয়ে পিছিয়ে ঠিক জায়গায় নিয়ে আসুন, টাইপ করুন যা চান, আর ডিলিট (Delete) মারলে কার্সরে থাকা অক্ষরটা মুছে যায়, ব্যাকস্পেস (Backspace সুইচ বা ← সুইচ) মারলে কার্সরের বাঁদিকের অক্ষরটা মুছে যায়। কখনো কখনো একটা কমান্ড হতে পারে বেধড়ক লম্বা, গ্নু-লিনাক্সে কমান্ড প্রম্পটে তিন চার লাইনের কমান্ড হরহামেশাই ঘটে, তখন ডান বা বাঁ তীর মেরে মেরে পৌঁছনোটা ঝামেলার। কমান্ডটার একদম গোড়ায় চলে যাওয়া যায় 'কন্ট্রোল-এ' (Ctrl-A) মেরে, একদম শেষে চলে যাওয়া যায় 'কন্ট্রোল-ই' (Ctrl-E) মেরে। কমান্ড লাইনের এরকম আরো অনেকগুলো তরকিব আছে, খুবই সুবিধেজনক, তবে সেগুলো কাজে লাগানোর আগে আর একটু কাজের জটিলতা বেড়ে ওঠা দরকার।

এই যে আপনার আদেশের ইতিহাসের এক হাজারি মনসবদারি, এর দলিলটা একটু শো করে নেব? পর্চা ম্যাপ, দাগ ম্যাপ এসব পরে লড়ানো যাবে, যখন আমরা কাস্টমাইজেশন নিয়ে কথা বলব, কাস্টমাইজেশন মানে আপনার কাস্টমের সঙ্গে প্রথার সঙ্গে মিলিয়ে আপনার মর্জিমাফিক করে নেওয়া, মানে আপনার পাঁঠা কোথায় কাটবেন, লেজে না শিঙে, লেজ ক-পিস, শিঙ কী সাইজের চোকলায়, মাংস কোথায় যাবে, বদরক্ত কোথায়, ইত্যাদি। এর মূল জায়গাটা হল এনভায়র্নমেন্ট ভ্যারিয়েবলগুলো, সেগুলো কী — মনে করতে পারছেন? কোথায় এসেছিল কথাটা? যাকগে, ঠিক

সময়ে মনে পড়ে যাবে। পরে আমরা এগুলোয় আসব, আমরা যদি এই জিএলটি পাঠমালায় সেই ভাবে নাও আসি, আপনাকে নিজে নিজে আসতেই হবে, পাঠমালা অর নো পাঠমালা, নয়তো আপনার সিস্টেম চিরকালই একটা ডিফল্ট পাঁঠা হয়ে থাকবে, আপনার ব্যক্তিগত একান্ত জনান্তিক পাঁঠা হয়ে উঠবে-না, চেঁচাবে একদম পপুলার ডেমোক্র্যাটিক হাটুরে ব্যাব্যারবে। পাঁঠা না গাধা? আমরা তো শূন্য নম্বর দিনে সিস্টেমকে গাধা বলে ডেকেছিলাম? আপনার হোম সুইট হোম ডিরেক্টরিতে দাঁড়িয়ে একবার '`ls`' মারুন। একটু জোরে মারুন, মানে জোরে কী-বোর্ডে মারবেন না, একটু জোরালো এলএস মারুন — '`ls -a`', যাতে ডটানন ফাইলগুলোকেও দেখায়, ডটানন ফাইলগুলোকে মনে আছে, ছয় নম্বর দিনের এক নম্বর সেকশনে? যাদের গোড়াতেই একটা ডট মানে বিন্দু মানে '.' থাকে, যাদের এমনিতে সিস্টেম দেখায়না? ডটানন কিন্তু শুধু ফাইল না, গোটা গোটা ডিরেক্টরি অব্দি ডট মুখে বসে থাকে। আমার নিজের সিস্টেমে একবার মেরে শো করে দি, দাঁড়ান।

.AbiSuite	**.ICEauthority**	.X.err	.Xauthority	.Xdefaults
.Xmodmap	.Xresources	**.adobe**	.bash_history	.bashrc
.blackboxrc	.cdrdao	.dvipsrc	.emacs	**.emacs.d**
.esd_auth	.exrc	**.fltk**	**.fonts**	.fonts.cache-1
.fonts.conf	**.gconf**	**.gconfd**	**.gnome**	**.gnome2**
.gnu-emacs	.gtkrc-kde	**.kde**	.kderc	.kermrc
.links	.mailcap	**.mcop**	.mcoprc	.mime.types
.mozilla	**.mplayer**	.muttrc	.padminrc	.profile
.qt	.recently-used	.skel	.sversionrc	**.thumbnails**
.urlview	.viminfo	**.wine**	.wmrc	.xcoralrc
.xemacs	.xim	.xinitrc	**.xmms**	.xserverrc.secure
.xsession	.xsession-errors	.xtalkrc	.y2log	**.yast2**

এই তালিকাটা একটু বদলে নিলাম, দেখানোর সুবিধের জন্যে, আর, '`ls -a`' সব ফাইল আর ডিরেক্টরিগুলোকেই দেখিয়েছিল, তার থেকে আমি শুধু ডটানন ফাইল আর ডিরেক্টরিগুলোকে বাদ দিয়ে দিলাম, যেগুলো আপনি এমনিতেই '`ls`' মেরে দেখতে পেতেন। আর এর মধ্যে ডিরেক্টরিগুলোকে আমি ব্লক করে দিলাম বোঝার সুবিধের জন্যে।

এবার, প্রশ্ন আপনার আদেশের ইতিহাস ধরে রাখার গল্পটা থাকছে কোথায়? এর জন্যে একটা কমান্ড লাগে, যার নাম 'set'। এই '`set`' একবার মেরে তারপর এন্টার মেরে দেখুন, আবার সেই শচারেক লাইন হুড়হুড় করে বেরিয়ে যাওয়ার গল্প। যেমন আমি সেট মেরে দেখলাম, আমার সেট খুব ছোটো, মাত্র তিনশো বিরাশি লাইনের, আপনার সেট কত লাইনের — আপনি গুনতে জানেন তো? না জানলে, এই পাঠমালাটা আর একবার গোড়া থেকে পড়তে পড়তে আসুন, দেখুন কোথাও আছে কিনা? এত বাজে বকেছি কনস্ট্যান্টলি, গোটাটা একটা শব্দকল্পদ্রুমের মত হয়ে গেছে, কী আছে আর কী নেই, অনেকটা আমার লেখাপড়ার টেবিলের মত, যাতে নাকি কু-লোকে বলে, হাতি হারালেও খুঁজে পাওয়া যাবেনা। একদমই বাজে কথা, আজ পর্যন্ত এত বছরে কোনোদিন কোনো হাতি ওখানে হারায়নি। এখানে সেই সেট মেরে পাওয়া শশ-লাইনের কয়েকটা মাত্র তুলে দেওয়া যাক, দেখুন তো বুঝতে পারেন কিনা। মনে রাখবেন, এটা কিন্তু আমি, ইউজার 'dd', সেট মেরেছিলাম আমার সুজে অপারেটিং সিস্টেমে, আর কোনো একজন ইউজার সেট মারলে কিন্তু অন্য একটা টেক্সট পাওয়া যেত।

```
BASH=/bin/bash
HISTFILE=/home/dd/.bash_history
HISTSIZE=1000
HOME=/home/dd
PATH=/home/dd/bin:/usr/local/bin:/usr/bin:/usr/X11R6/bin:/bin:/usr/games:/opt/gnome2/bin:/opt/gno
    me/bin:/opt/kde3/bin:/usr/lib/java/jre/bin:/opt/gnome/bin
INFOPATH=/usr/local/info:/usr/share/info:/usr/info
MANPATH=/usr/local/man:/usr/share/man:/usr/X11R6/man:/opt/gnome2/man:/opt/gnome/man
LANG=en_US
LOGNAME=dd
OSTYPE=linux
```

PS1='\u@\h:\w> '
PWD=/home/dd
SHELL=/bin/bash

এর দ্বিতীয় লাইন দেখুন, ‘HISTFILE=/home/dd/.bash_history’, এর মানে, আদেশের ইতিহাসটা রয়ে যায় ওই ‘/home/dd/.bash_history’ ফাইলে। আর এর এক হাজারি মনসবদারির আয়তনটা লেখা আছে এখানে তৃতীয় লাইনে ‘HISTSIZE=1000’। এবার একটা জিনিষ খেয়াল করুন, এর প্রতিটা লাইনে একদম বাঁদিকে একটা কথা লেখা, ক্যাপিটাল কেস বা বড় হাতের ল্যাটিন বর্ণমালায়, তার পর একটা সমান-চিহ্ন, তারপর ডানদিকে একটা কথা লেখা ছোট হাতের বর্ণমালায়। কাঠামোটা খেয়াল করুন, বাঁদিকের ওই বড় হাতের লেখা কথাগুলোই হল সিস্টেম ভ্যারিয়েবল। ভ্যারিয়েবল বলতে বোঝায় চলরাশি, যা চলে বেড়ায়, বদলে যায় এক অবস্থা থেকে আর এক অবস্থায়, এক জন থেকে আর এক জনে, এক সিস্টেম থেকে আর এক সিস্টেমে। এর উল্টো হল কন্সটান্ট বা স্থিররাশি। যা কখনো বদলায় না। এই শব্দগুলো পাওয়া যায় গণিতে, পদার্থবিদ্যায়, অর্থনীতিতে, যে কোনো বিজ্ঞানে। আমাদের বাস্তব পৃথিবীতেও চলরাশি আর স্থিররাশি আছে, যেমন বৃত্তের ক্ষেত্রফল বার করার সময় যে ‘π’ বা ‘পাই’-কে পাই আমরা, সেটা একটা স্থিররাশি বা কন্সটান্ট। কোনো অবস্থায় কোনো জায়গায় এর মান বদলাবে না। ইচ্ছে হলে একদিন বিকেল বিকেল বৃহস্পতিতে বা সকাল সকাল শনিতে গিয়ে একটা বৃত্ত মেপে দেখুন, সেখানেও এর মান সেই একই হবে। কিন্তু অন্য অনেককিছু আবার ভ্যারিয়েবল বা চলরাশি। আমাদের কার কত টাকা আছে এটা মানুষ থেকে মানুষে বদলায়, তাই ভ্যারিয়েবল। একটা বইয়ের চ্যাপ্টারে চ্যাপ্টারে কত করে শব্দ আছে সেটা একটা চলরাশি। আবার কটা আদেশ অব্দি সিস্টেম মনে রেখে দেবে, এটাও একটা ভ্যারিয়েবল। এখানে বলা আছে যে এর সাইজ ‘HISTSIZE’ হবে একহাজার। শেলের খোঁজার পাথ বা পথনির্দেশ ব্যাপারটা মনে পড়ছে, পাঁচ নম্বর দিনের আলোচনা থেকে? ‘dd’ নামের কোনো ব্যবহারকারী কোনো আদেশ দেওয়ামাত্র যেখানে যেখানে শেল খুঁজে দেখবে, ওই আদেশটার মানে কী, সেই পথনির্দেশটা দেখুন দেওয়া আছে, ‘PATH’ নামের ভ্যারিয়েবল। অন্য নামের ভ্যারিয়েবলগুলো নিজেই আন্দাজ করার চেষ্টা করুন। প্রথম লাইনেই দেওয়া আছে, আমার ‘BASH’ নামের সফটওয়ারটা, যা আমার শেল, সেটাকে কোথায় পাওয়া যাবে। আবার আমার শেল যে ওই ব্যাশই, সেটা দেওয়া আছে এখানে তোলা লাইনগুলোর শেষ লাইনে, ‘SHELL’ নামের ভ্যারিয়েবল। ‘PS1’ নামের ভ্যারিয়েবলটা আসলে আমার কমান্ড প্রম্পট কেমন দেখতে হবে সেটা বলে দিচ্ছে। ডানদিকেরটা একটা ছক বা টেমপ্লেট, যেখানে মানগুলো শেল নিজেই ভরে নেবে, শুধু একটা জিনিষ মনে রাখুন, ‘\’ চিহ্নটা আসলে বলে দেয় শেলকে যে এর পরের কথাটাকে তুমি আক্ষরিকভাবে নাও। পরে বুঝে যাবেন পুরোটা। এখানে তোলা লাইনগুলোর দ্বিতীয় লাইনে লেখা ‘/home/dd/.bash_history’ হল আমার বা ‘dd’ নামের ইউজারের আদেশের ইতিহাস লিখে রাখার জায়গা। ‘`less /home/dd/.bash_history`’ কমান্ড দিয়ে আমি পাব আমার শেষ ব্যবহার করা একহাজারখানা আদেশের তালিকা। আপনি পাবেন আপনার হোমের ‘.bash_history’ ফাইলে।

এবার প্রশ্ন হল, ভ্যারিয়েবলগুলোর মান যে এটাই এটা শেলকে বলে দিল কে? আমাকে তো শেল কোনোদিন ফোন করেছে বলেও মনে পড়ে-না? তাহলে? নিজেই খুঁজে দেখুন তো? একটু মনে করিয়ে দিই। এই ফলাফলটা পাওয়া গেছিল ‘set’ কমান্ড দিয়ে। তার মানে আপনি ‘`man -k set`’ কমান্ড লাগাবেন, ছ নম্বর দিনের থেকে মনে করুন। বা সরাসরি ‘`man set`’ কমান্ড-ও দিতে পারেন। এই কমান্ড দিয়ে যা পাবেন সেটা হল ‘bash’-এর ম্যানুয়াল পেজ, মানে ‘`man bash`’ দিয়ে যা পেতেন, কারণ, এই ‘set’ হল সরাসরি ‘bash’-এর কাজ। এবার ম্যান পেজ তো পেলেন, পড়ে ফেলুন, একাধিকবার। এই বিশেষ সমস্যাটায় আপনি খুঁজছেন কনফিগারেশন ফাইলগুলোর নাম, তো গুড, একটু চালিয়াতি খেলার চেষ্টা করা যাক, আপনি তো জানেন ম্যান পেজে বা লেসে কী করে খুঁজতে হয় একটা বিশেষ শব্দবন্ধ, ‘\’ টাইপ করুন, দেখুন একদম নিচে সেটা ফুটে উঠেছে, এবার টাইপ করে দিন ‘file’ শব্দটা। এন্টার মারুন। প্রথম যেখানে ‘file’ আছে সেখানে আপনাকে নিয়ে গেল, ঠিক জিনিষটা না-পেলে আবার একবার ‘\’ মারুন। এবার আর কিছু টাইপ না-করেই এন্টার মারুন। ও আপনাকে দ্বিতীয় যেখানে ‘file’ সেখানে নিয়ে গেল। না-পেলে আবার করুন। আর একটু হেল্প করব? আপনি পঞ্চমবার ‘file’ যেখানে পাবেন, সেখানেই পেয়ে যাবেন এই তথ্যটা — ‘`set`’ মেরে আমি যে তালিকাটা পেয়েছিলাম, যার কয়েক লাইন তুলে দিলাম, সেই তালিকাটা ‘bash’ পাচ্ছে এই চারটে ফাইল থেকে, ‘/etc/profile’, ‘~/.bash_profile’, ‘~/.bash_login’ এবং ‘~/.profile’। একটু আগে যে ডটানন

ফাইল আর ডিরেক্টরির একটা টেবিল আমরা বানিয়েছিলাম, তাতে কি এদের কারোর নাম আপনি পেয়েছিলেন? শুধু একটাই জিনিষ, এই তেরাবেঁকা '~' চিহ্নটা কী বস্তু? আমরা যে যেরকম ঘরেই থাকি-না কেন, দশ বাই দশ, চোদ্দ বাই চোদ্দ, যাই হোক, তার দেওয়ালগুলো যতই সোজাসোজা হোক, এই তেরাবেঁকা ত্রিভঙ্গমুরারি টিল্ড-ই হল আপনার হোম, আমার হোম, সবার হোম। যখনই কেউ '~' টাইপ করে, সিস্টেম জানে সে কে, বুঝে নেয় যে এটা তার হোম, যেমন আমার বা 'dd' নামের ইউজারের বেলায় এই '~' মানে '/home/dd/'। 'atithi' ইউজার হলে '~' মানে '/home/atithi/'।

আদেশের ইতিহাসের সূত্রে একটু সিস্টেম কনফিগারেশন জানা হল আমাদের, পরে আবার আসব আমরা এইসব কথায়, এখন ফিরে যাওয়া যাক আমাদের লিংক নিয়ে আলোচনায়।

৯.৪।। হার্ড লিংক

এখন আমরা জানি কী করে আগের আদেশ ফেরত আনে সিস্টেম, এবার, সেই ফেরত আনা আদেশ দিয়ে আর একবার আমরা '`ls -al`' কমান্ডটা মারব এই 'onefile', 'twofile', আর 'threefile'-এর উপর। এবার আর আমরা '`ls -ail`' মারবনা, কারণ এই তিনটে ফাইলের আইনোড নম্বরের চক্করটা তো আমরা ইতিমধ্যেই জানি। এখন আমাদের 'onefile' কিন্তু বদলে গেছে, একটু আগেই আমরা তাতে টেক্সট ভরেছি '`onefile is one file`' — এই সমস্ত চিহ্নগুলো পরপর, যার মধ্যে ষোলটা অক্ষর আছে, ফাঁকা জায়গা বা স্পেস আছে তিনটে, এবং তারপর একটা অদৃশ্য চিহ্ন, যার নাম নিউলাইন বা নতুন-লাইন। '`ls -al`' মেরে দেখা যাক।

```
-rw-r--r--   2  dd  users    20 2003-12-24 10:51 onefile
lrwxrwxrwx   1  dd  users     7 2003-12-24 10:50 twofile -> onefile
-rw-r--r--   2  dd  users    20 2003-12-24 10:51 threefile
```

এবার দেখুন 'onefile' আর 'threefile' দুটোরই বাইটসাইজ দেখাচ্ছে কুড়ি করে, আর বেচারা 'twofile', তার কোনো বদল হলনা গা, এটা কি একটা বিচার হল, যার সঙ্গে সে লিংকিত, নয় একটু কোমলতার সঙ্গেই, সফট লিংক না, সেই 'onefile' ফনফন করে বেড়ে উঠল শূন্য থেকে কুড়ি, ফুটপাথ ফাটিয়ে মাথা-গজানো দুর্বোঘাসের মত যারা বারবার প্রমাণ করে, স্টিল স্প্রিং ইজ স্প্রিং ইভন ইন দি টাউন। তার সঙ্গে একই ভাবে বেড়ে গেল ওই আর একটা লিংক 'threefile', নয় একটু হার্ড। আর সে 'twofile' সেই সাত এক্কে সাতের নামতা? দেখুন তো কুড়ি সাইজটা কি আপনার চেনা লাগছে? ষোল আর তিন আর এক যোগ করে কত হয় যেন? এবার দেখুন তো, ওই সাতের রহস্যটার কোনো মানে বুঝছেন? ওই সংখ্যাটাতেই, সফট লিংকের বাইটসাইজেই, লেখা নেই তো গরবিনী নরম-লিংকিনীর প্রাণের কানুর নাম? '`onefile`' নামে কটা অক্ষর? একটু আগে ৯.১ সেকশনে আমি আপনার বানানো সফট-লিংক ফাইলের সাইজ বলে দিচ্ছিলাম মূল ফাইলের নামটার অক্ষর গুনে। প্রত্যেকটা উদাহরণেই গুনে দেখুন। সফট লিংকের বুকে শুধু লেখা থাকে তার নাম যার সে লিংক।

আর 'threefile' কেন বাইটসাইজ কুড়ি সেটা তো আপনারা আগেই জানেন, এটা হল 'onefile'-এর একটা হুবহু প্রতিরূপ। যে কারণে, লিংকসংখ্যার জায়গায়, 'onefile' আর 'threefile' দুজনেরই লেখা আছে দুই। কোনো দিক দিয়েই 'onefile' আর 'threefile'-এর ভিতর কোনোরকম কোনো পার্থক্য করা যায়না, এরা দুজনে তো আসলে একটাই ফাইল, শুধু দুটো আলাদা নাম দিয়ে আমি বোঝাচ্ছি একটাই ফাইলকে। যে কারণে তাদের আইনোড নম্বর এক, সেই আইনোড নম্বর যা দিয়ে সিস্টেম একটা ফাইলকে বোঝে, যাকে আমরা বুঝি তার নাম দিয়ে। তাই আমরা যাকে 'onefile' আর 'threefile' এই দুই নাম দিয়ে বুঝছি সে আসলে একটাই আইনোড নম্বর, সিস্টেমের কাছে একটাই ফাইল। দুই আলাদা নামে তাকে রাখার ফলে ঘটেছে এটাই যে একটাকে উড়িয়ে দিলেও অন্য নামটা রয়ে যাবে, তাই আইনোড নম্বরটাও, তাই ফাইলটাও, সিস্টেম তাকে ওড়াতে পারবেনা, যতক্ষণ না তার শেষ হার্ড লিংকটাও উড়িয়ে দিচ্ছি আমরা। মূল ফাইল 'onefile'-কে উড়িয়ে দিয়ে দেখুন, আর সফটলিংক 'twofile' ফাইলকে ক্যাট করে আপনি দেখতে পাবেন না, সিস্টেম আপনাকে জানাবে, মূল ফাইল 'onefile' আর নেই। কিন্তু 'onefile' উড়িয়ে দিয়েও '`cat threefile`' করুন, দেখবেন অবিকল মূল ফাইলটা আপনি পেয়ে যাচ্ছেন।

যেসব পরিস্থিতিতে আমাদের লিংক ফাইল ব্যবহার করতে হয় তার কোথাও কোথাও সফট লিংক করা যায়না, হার্ড লিংক করতেই হয়, তার কারণ, অনেক প্রোগ্রামই আছে যারা সিম্বলিক লিংক বোঝেনা। ধরুন কপি করার আমাদের

যে কমান্ড ‘cp’। এই ‘cp’ কোনো সিম্বলিক লিংকের উপর ব্যবহার করলে সে সিম্বলিক লিংকের লিংকত্বটাই বোঝেনা, নিট ফাঁপা নামাবলীটাই কপি করে আনে, মূল ফাইল কপি হয়না। এটা হার্ড লিংকের বেলায় হওয়ার কোনো চান্সই নেই। হার্ড লিংক তো জাস্ট আর একটা নাম, আর নাম যাই হোক, সে তো বাস্তব একটা ফাইল, আর বাস্তব তথ্য নিয়ে বাস্তব একটা ফাইল তো কপি হবেই। তবে ‘cp’ কমান্ডটাকেও সিম্বলিক লিংক নিয়ে কাজ করানো যায়, সেই রকম অপশান আছে। . . . *খেজুরগাছে . . .*

কিন্তু হার্ড লিংকের কঠোরতায় কখনো কখনো ব্যথাও লাগে যখন জানা যায় এই কঠিন যোগাযোগ কখনো দুটো আলাদা ফাইলসিস্টেমের মধ্যে ঘটতে পারেনা। মূল ফাইল আর লিংক দুটোকেই একই ফাইলসিস্টেমে থাকতে হবে। দেখেছেন, গরু তার শিঙে যা নিয়েই ঘুরুক, তার পথ শেষ হয় শ্মশানে, আমরা ফের ফেরত এসেছি ফাইলসিস্টেমে, যেখান থেকে, আমরা হার্ড ডিস্ক পার্টিশন আর ব্লকের কাহিনীতে যাওয়ার আগে, আগের ছ নম্বর দিনে, ডিভাইস, ডিভাইস ফাইল আর নাম নিয়ে আলোচনা শেষ করার পর, অন্য কিছু কথা বলে নেওয়ার দরকারে আজ শুরু থেকেই চলে গেছিলাম ফাইলের অনুমতি মালিকানায়, তারপর লিংকে। মধ্যে একটু কমান্ডের ইতিহাস তথা ব্যাশ-এর কনফিগারেশন ফাইল নিয়েও কথা বলে নিয়েছি। এবার ফেরত আসা যাক আমাদের মূল আলোচনায়, মানে ফাইলসিস্টেমে।

১০।। হার্ডডিস্ক, পার্টিশন

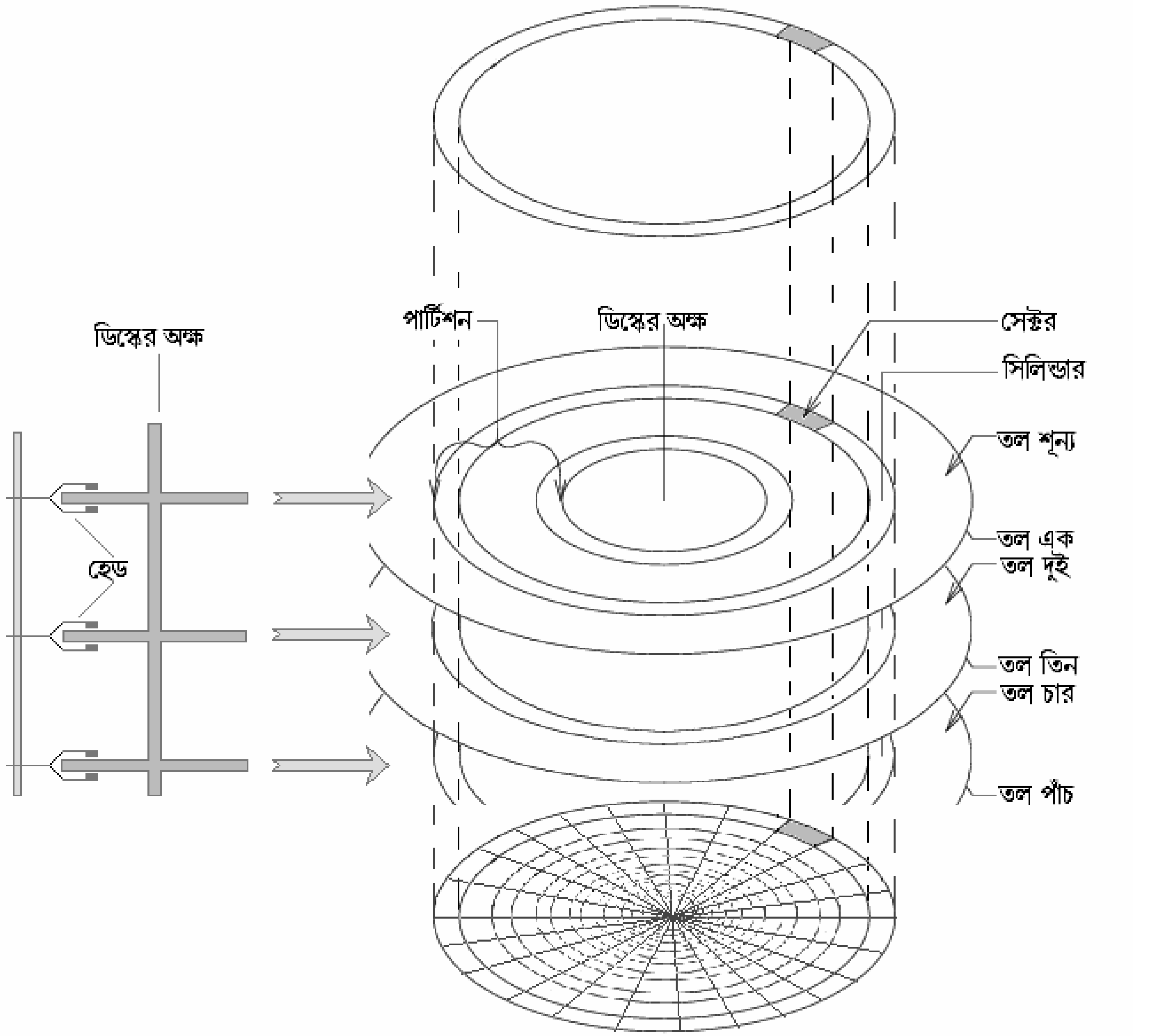
শূন্য নম্বর দিনে হার্ড ডিস্কের ভিতরে প্ল্যাটার আর হেডগুলোর কাজ করার কায়দাটা মনে আছে? একটা ডিস্ক মানেই কয়েকটা করে প্ল্যাটার। থালার সোজা আর উল্টো আছে, কিন্তু প্ল্যাটারের দুটো দিকই সোজা দিক, তাই প্ল্যাটার পিছু দুটো করে তল যেখানে চৌম্বক পদার্থের উপর তথ্য লেখা যায় চৌম্বক পদার্থের উপর লেখার কায়দা সম্পন্ন ঘুরতে এবং নড়তে থাকা হেড দিয়ে, মনে পড়ছে? এবারে সেটা একটু ভালো করে বুঝি। সেখান থেকে আমরা পার্টিশন এবং ফাইলসিস্টেম বোঝার দিকে যাব। ছয় নম্বর দিন থেকে যে প্রসঙ্গটা শুরু হয়েছে এবার আমরা তার ফিনিশের দিকে যাচ্ছি, ছয়ের আলোচনাগুলো একটু মাথায় এনে নিন। ফাইল, ফাইলনাম, ফাইলের রকম। ডিরেক্টরিসিস্টেম বা ডিরেক্টরিব্যবস্থা। মানে সিঁড়িভাঙার মত করে স্তরে ডিরেক্টরি এবং ফাইলের রাখার বন্দোবস্ত। বহুসময় চালুকথায় একেও আমরা ‘ফাইলসিস্টেম’ বলি। তবে, এটা স্বীকৃত ব্যবহার নয়। ‘ফাইলসিস্টেম’ শব্দটার স্বীকৃত এক নম্বর ব্যবহার মানে হল গোটা ঐক্যবদ্ধ ফাইল এবং ডিরেক্টরির সমগ্রতা। ‘ফাইলসিস্টেম’ শব্দটার স্বীকৃত দু-নম্বর ব্যবহার হল ফাইল বানানো পড়া লেখা এবং রাখার নানা রকম — যে আলোচনায় আমরা এখনো আসিনি, শুধু এদের দুচারটে নামের উদাহরণ দেওয়া ছাড়া, এক্সএফএস, ইএক্সটিটু, ফ্যাটথার্টিটু, ইত্যাদি। এক নম্বর ফাইলসিস্টেম আলোচনার সূত্রে এল পার্টিশন এবং ডিভাইস নিয়ে খুব প্রাথমিক কিছু কথা। তারপর ডিভাইস, ডিভাইস ফাইল, তারপর এল ফাইলের অনুমতি এবং অধিকারের আলোচনা। তারপর ফাইল লিংক। তারপর আমরা মাইল্ডলি একটু কুপথে গেছিলাম, ডাইগ্রেস করেছিলাম ব্যাশ এবং তার কনফিগারেশনের প্রসঙ্গে। এবার আবার আমরা ফেরত আসছি ফাইলসিস্টেমের কথায়, পার্টিশনটাকে বুঝে আমরা যাব ফাইলসিস্টেমের দু-নম্বর ব্যবহারটা নিয়ে দুচার কথায়, তারপর শেষ করব গ্নু-লিনাক্সের ফাইলসিস্টেম হায়েরার্কিব্যবস্থার স্বতন্ত্র স্বীকৃত এবং গৃহীত আকার নিয়ে কথায়।

১০.১।। হার্ডডিস্কের ভৌত জ্যামিতি — সেক্টর, ট্র্যাক, সিলিন্ডার, পার্টিশন

এখানে এই ছবিতে আমরা তিনটে প্ল্যাটার দেখিয়েছি, তার মানে তিন দুগুনে ছ-খানা তথ্য লেখার তল, প্রতি তলে তথ্য লেখার জন্যে একখানা করে হেড, তার মানে ছ-খানা হেড। শুধু তলগুলোকে আমরা এক থেকে ছয় না-গুনে, গুনেছি শূন্য থেকে পাঁচ, সেটাই প্রথা। দেখুন দেখানো আছে পরপর। ছটা তলে মোট ছটা হেড একই তালে নড়ে, একই সঙ্গে, একা একা আলাদা করে নাড়ানো যায়না। তার মানে, খেয়াল করুন, যে কোনো একটা নড়াচড়া মানেই ছ-খানা হেডের একই সাথে নড়া, একই ভাবে, একই গতিতে, ছটা আলাদা আলাদা তলের উপর। এদের এই গতির ঐক্য থেকেই গড়ে উঠেছে হার্ডডিস্কের ভৌত ভূমিকে বোঝার আমাদের একক, যার নাম সিলিন্ডার।

থ্রি-অ্যান্ড-হাফ-ইয়ার্ডস-ই তো নাম, সেই গল্পটার কথা মনে করুন, যতটা জমি একসঙ্গে তুমি দৌড়তে পারবে, ততটা জমিই তোমার — লোকটা তাতে এমন দৌড় দৌড়লো যে শেষ অব্দি দৌড়তে দৌড়তে জমিতেই পড়ে গেল, অত অত জমির কিছুই লাগল না, লাগল মাত্র থ্রি-অ্যান্ড-হাফ-ইয়ার্ডস, কত জমি আর লাগবে, যত বড় দৌড়বাজেরই হোক,

লাশ তো। হার্ড ডিস্কের বেলায় শুধু এই একা একলা একটা দৌড়বাজকে রিপ্লেস করে দিন ছ-খানা দৌড়বাজ হেড দিয়ে। মালেরা দৌড়চ্ছেও একই তালে, এই ছ-খানা হেডের কমিউন। প্রত্যেকে যতটা করে দৌড়চ্ছে তার পরিমাণ সমান — সেই পরিমাণের ছয়গুণ হল কমিউনের মোট জমি। একটা হেডের একটা দৌড়ের ছ-খানা প্রতিরূপ তৈরি হচ্ছে তিনটে প্ল্যাটারের ছটা তলে ছখানা হেডের ছটা দৌড়ে। আমরা এখানে তিনটে প্ল্যাটার দেখিয়েছি বলে ছজনের কমিউন, প্ল্যাটার বাড়া মানেই দৌড়বীর বাড়া, সেই একই হারে বেড়ে যাওয়া কমিউনের জমি। দেখুন, এই যে কমিউনের জমির ধারণাটা তৈরি হয়ে গেল, এটাই আমাদের কাজে লাগবে। এই জমিটা ঠিক কীরকম হবে? প্রতিটা প্ল্যাটারের প্রতিটা তলে একই ভৌগোলিক সংস্থানে থাকবে এই জমি। যেমন দেখুন একটা তলে একটা হেডের এক সেক্টর জমির ঠিক ছ-খানা প্রতিরূপ থাকবে ছখানা তলে। এর ছটা টুকরো জমিকে মিলিয়ে তৈরি হবে কমিউনের জমির একটা সেক্টর। মাইরি, ছবিটা দেখে বোঝার চেষ্টা করুন, হাউ-টু ডকুমেন্টের একটা ছবির একটু ব্যবহার করে, অনেক পরিশ্রমে, কাল বড়দিনের রাত্তিরে পৌনে বারোটা অব্দি, আজ আবার ভোরে সোয়া চারটে থেকে প্রায় তিনঘন্টা গাবিয়ে ছবিটা পাকানো।



এক একটা তল মানে একগুচ্ছ কনসেন্ট্রিক বা সমকেন্দ্রিক বৃত্ত, ছবির একদম নিচে দেখুন, একটা তলকে আমরা দেখিয়েছি। এই সমকেন্দ্রিক বৃত্তগুলো হল ট্র্যাক, পরপর ক্রমানুসারী সংখ্যা দিয়ে চিহ্নিত। শূন্য নম্বর তলে ঠিক যেখানে আছে এক নম্বর ট্র্যাক, তার ঠিক খাড়াখাড়ি উপরে বা নিচে রয়েছে অন্য প্রতিটা তলেরই এক নম্বর ট্র্যাক। প্রতিটি নম্বরের ট্র্যাকের বেলাতেই এটা সত্যি। তার মানে, প্রতি নম্বরের ট্র্যাক আছে ছ-খানা, ঠিক খাড়াখাড়ি উপরে-নিচে, তিনটে প্ল্যাটারের ছ-টা তলে। যদি প্ল্যাটারের সংখ্যা বেশি হয়, প্রতি কমিউনে একই নম্বরের ট্র্যাকও বাড়বে।

প্ল্যাটার চারটে হলে একই নম্বরের আটটা ট্র্যাক থাকবে এক একটা কমিউনে, পাঁচটা প্ল্যাটার থাকলে দশটা — এই ভাবে সংখ্যাটা বাড়বে। এবার খাড়াখাড়ি প্ল্যাটারগুলোকে পরপর একই সঙ্গে ভাবুন, যেভাবে আমরা ছবিতে ভাঙা দাগ দিয়ে দেখিয়েছি, প্রত্যেকটা প্ল্যাটারের প্রতিটি একই নম্বরের ট্র্যাককে একসঙ্গে নিয়ে এক একটা কমিউনকে আমরা একটা খাড়া সিলিন্ডার বা চোঙ আকারে ভাবতে পারি। এক একটা সিলিন্ডার তৈরি হচ্ছে ঠিক যেকটা প্ল্যাটার আছে তার দ্বিগুণ সেক্টর নিয়ে, আমাদের ছবির বেলায় ছটা। একটা সেক্টর বেয়ে একটা হেড নড়ছে মানে গোটা সিলিন্ডার বেয়ে ছটা হেড নড়ছে। মাথায় রাখবেন, এটা কিন্তু একটা ভৌতিক চোঙ, আমরা আমাদের চিন্তায়, বুঝবার সুবিধের জন্যে বানিয়ে তুলছি। বাস্তব হার্ডডিস্কটা আদৌ এরকম চোঙদার নয়।

এবার আমরা গোটা হার্ডডিস্কটাকে ভেবে ফেলতে পারি এইরকম কিছু সিলিন্ডারের সমাহার হিশেবে। মোট কটা সিলিন্ডার তাহলে হতে পারে একটা হার্ডডিস্কে? ঠিক যেকটা ট্র্যাক বা সমকেন্দ্রিক বৃত্ত আছে এক একটা তলে সেইকটাই সিলিন্ডারের সমাহার বলে ভাবা যাবে হার্ডডিস্কটাকে। গোটা ডিস্কটা কিন্তু ঘুরছে একই সঙ্গে, আমার মেশিনের হার্ডডিস্কের বেলায় মিনিটে পাঁচ হাজার চারশো বার করে, আরো দামি হার্ডডিস্কের বেলায় গতিটা আরো বেশি। হার্ডডিস্কটা ঘোরার মানে ঘুরছে তার মধ্যে ছখানা তল, আর তাদের উপর দিয়ে নড়ছে ছখানা হেড, একই সঙ্গে ছজন ছটা তলে ছটা আলাদা আলাদা ট্র্যাক বেয়ে, যে প্রত্যেকটা ট্র্যাকেরই তার নিজের তলে একই ক্রমিক নম্বর। প্রতিটা ট্র্যাককে আবার ভাবা হয় ছোট ছোট টুকরোয় যাদের নাম সেক্টর। ছবিতে দেখুন আমরা একটা সেক্টরকে দেখিয়েছি। সচরাচর এক সেক্টর মানে পাঁচশো বারো বাইট তথ্য। তার মানে গোটা হার্ডডিস্কের তথ্য রাখার ক্ষমতা দাঁড়ালো — (পাঁচশো বারো) × (প্রতি ট্র্যাকে সেক্টরের সংখ্যা) × (প্রতি তলে ট্র্যাকের সংখ্যা) × (তলের সংখ্যা) = (পাঁচশো বারো) × (প্রতি ট্র্যাকে সেক্টরের সংখ্যা) × (প্রতি তলে ট্র্যাকের সংখ্যা) × (প্ল্যাটারের সংখ্যা × দুই)। সচরাচর আমরা ডিস্ক নিয়ে কথা বলি যখন, এই সিলিন্ডার আর সেক্টরের নিরিখেই বলি। তল, ট্র্যাক এইসব রোজকার কথায় আসেনা।

ছয় নম্বর দিনে যাদের নিয়ে আমরা অনেক কথা বলেছি তাদের ফাইলসিস্টেমের আকার বুঝতে গিয়ে, হার্ডডিস্কের পার্টিশনগুলোকে আবার ভাগ করা হয় এই সিলিন্ডারের সীমানা বরাবর। প্রথম দিকে, ইনস্টলেশনের সময়, অনেককেই দেখেছি চটে যেতে, কেন আমি ঠিক যে সাইজের পার্টিশন হবেনা? যেটা হচ্ছে সেটা দেওয়া ফিগারটার খুব কাছাকাছি, কিন্তু হুবহু সেইটা হবেনা কেন? হয়না ঠিক এই জন্যে। আমি যত মেগাবাইটের পার্টিশন চাইছি, ঠিক সেটা করতে গেলে হয়ত একটা সিলিন্ডারকে ভেঙে ফেলতে হচ্ছে। একটা সিলিন্ডারে ঠিক যতটা তথ্য আসতে পারে, হিশেব করে বার করে নিন, এবার তার কোনো সঠিক মাল্টিপল বা গুণিতকে দিন আপনার কাঙ্খার পার্টিশনের আকার, দেখবেন একদম খাপে খাপে সেটাই করছে। অর্থাৎ প্রতিটি পার্টিশন শুরু হবে হুবহু একটা সিলিন্ডারে, আবার শেষ হবে হুবহু একটা সিলিন্ডারে। পার্টিশনটার ভিতরকার আর বাইরেকার পরিধি, ছবিটার সঙ্গে মিলিয়ে বোঝার চেষ্টা করুন। একটা নির্দিষ্ট পূর্ণাঙ্গ সিলিন্ডারের বৃত্তে শুরু হবে, গোটাটা নিয়ে, আবার শেষ হবে আর একটা নির্দিষ্ট পূর্ণাঙ্গ সিলিন্ডারের বৃত্তে, গোটাটাকে নিয়ে।

কিন্তু এখানে মৃদু ঘাপলা আছে একটা, পার্টিশনের সিলিন্ডারগুলোর মাপজোক রাখার জন্যে যে পার্টিশন টেবিল, তাতে জায়গা থাকে মাত্র বল লিখে রাখার জন্যে জায়গা থাকে মাত্র দশ বিট। দশ বিট জায়গা মানে কতটা তথ্য রাখা যায়, হিশেব করে বলুন তো? যদি না-পারেন, তো মনে করে বলুন, শূন্য নম্বর দিনে গাণিতিক তথ্যের মাপজোকের সূত্রে আমরা আলোচনা করেছি, না-পারলে দেখে নিন, মোট তথ্য রাখার সীমা ১০২৪। মানে ১০২৪-টা অব্দি আলাদা আলাদা সম্ভাবনাকে হাজির করা যায়। তাই, একটা পার্টিশনেরও সিলিন্ডার সংখ্যার চূড়ান্ত সীমা ওই ১০২৪। এই ঝামেলার জন্যে হার্ডডিস্কের সিলিন্ডারের চূড়ান্ত সীমা হত ১০২৪। এই সমস্যার সমাধান হয়েছে এলবিএ (LBA — **L**ogical/**L**arge/**L**inear-**B**lock-**A**ddressing/**A**rray) এসে। সেখানে এই ভৌত সিলিন্ডারের ধারণাটাকে বদলে নেওয়া হয় একটা যুক্তিনির্মিত বা লজিকাল ছকে। একটা একমাত্রিক ঋজুরেখ ব্লকের সারি বলে ভাবা হয় গোটা হার্ডডিস্কটাকে। ঠিক একটা সরলরেখায় যেমন পরপর বিন্দু আসতে থাকে, গতিটা ঘটতে পারে কেবল একটা মাত্রাতেই, রেখা বরাবর, সেরকম গোটা হার্ডডিস্কের ভৌত ধারণক্ষমতাটাকে ভেবে নেওয়া হয় পরপর ব্লকের একটা সারি হিশেবে, ডিস্কের ভৌত জ্যামিতিতে আর যাওয়াই হয়না। হার্ডডিস্কের ধারণকে ভাবার এককই হয়ে গেছে সিলিন্ডার, তাই এলবিএ ছকে সাজিয়ে নেওয়া ব্লকটেবিলও সিলিন্ডার দিয়েই দেখায়, কিন্তু, এই এলবিএ আকারে ছকে

ফেলা একটা হার্ডডিস্কের একটা সিলিন্ডারের শুরু বা শেষের সঙ্গে আদত ভৌত হার্ডডিস্কের একটা ভৌত সিলিন্ডারের শুরু বা শেষের কোনো সম্পর্কই নেই। বায়োস রম চিপে যে আদেশগুলো ভরা থাকে, যার কথা আমরা আলোচনা করেছি এক আর দুই নম্বর দিনে, সেই আদেশগুলোর মধ্যে একটা বিশেষ আদেশস্তর বাড়িয়ে নেওয়া হয়, যার নাম 'INT13H', যাতে সরাসরি বায়োস হার্ডডিস্কটাকে আর ভৌত ডিস্ক হিশেবে না-ভেবে, একটা যৌক্তিক ছকে আবদ্ধ ব্লকের সারি হিশেবে নিতে পারে। এতে হার্ডডিস্কের একটা ভৌত পার্টিশনের ১০২৪ সিলিন্ডারের সীমা, এবং স্মৃতি ধরে রাখার ৫২৮ মেগাবাইটের সীমা অতিক্রম করতে পারা যায়।

১০.২।। পার্টিশন — প্রাথমিক ধারণা

আমরা জানি, একটা হার্ডডিস্ককে ফর্মাট করা মানে কী — তাকে কাঁচা ভৌত চৌম্বক পদার্থের একটা সমাহার থেকে ফাইল তথা ডিরেক্টরিসিস্টেম বানানোর মত জায়গায় নিয়ে আসা। যেকোনো অপারেটিং সিস্টেমের মত গ্নু-লিনাক্স সিস্টেমেও একটা হার্ডডিস্ক ব্যবহার করার আগে তাকে ব্যবহারযোগ্যতায় আনতে হয়, মানে ফর্মাটিং করতে হয়। এই ফর্মাটিং করেই অপারেটিং সিস্টেম কাঁচা হার্ডডিস্কটাকে নিজের তথ্য রাখার এবং পড়ার উপযোগী করে তোলে, কোনো ট্র্যাকে ত্রুটি থাকলে সেটাকে চিহ্নিত করে, যাতে পরবর্তী সময়ে তথ্য লেখার বা পড়ার সময়ে তাদের বাদ দিয়ে রাখা যায়। এর প্রাথমিকতম ফর্মাটিং-টা করা অবস্থাতেই আইডিই বা স্কাসি হার্ডডিস্কগুলো আমাদের কাছে আসে। আমরা ফর্মাটিং বলতে সচরাচর যে ক্রিয়াটা করি তার মানে মূলত ফাইলসিস্টেম বানানো। এর প্রথম স্টেপ হল পার্টিশন বানানো বা গোটা হার্ডডিস্কটাকে ছোট ছোট টুকরোয় ভাগ করা। পার্টিশনটা তৈরি হয়ে যাওয়ার পর থেকে সেটাকে অন্য পার্টিশনগুলো থেকে স্বতন্ত্র এবং স্বাধীনভাবে ব্যবহার করা যাবে, যেন সেটা নিজেই আলাদা একটা হার্ডডিস্ক। ছ-নম্বর দিনের ৬ নম্বর সেকশনের ওই টেবিলটা মনে করুন, যেখানে একই সঙ্গে একই হার্ডডিস্ক '/dev/hda'-তে দুটো পার্টিশন '/dev/hda1' আর '/dev/hda5', সেখানে ফাইলসিস্টেম উইন্ডোজ ফ্যাটথার্টিটু আর '/dev/hda6' হল তিন নম্বর পার্টিশন যেখানে ফাইলসিস্টেম রাইজারএফএস। এবং ভাবুন, দুটো আলাদা অপারেটিং সিস্টেম এদের ডিরেক্টরি এবং ফাইলকাঠামো তৈরি করেছে, প্রথম দুটোর বেলায় উইন্ডোজ আর তৃতীয়টার বেলায় স্ল্যাকওয়ার। এবং যখন কোনো একটা অপারেটিং সিস্টেমে বুট করা হয় তখন উইন্ডোজ তার বুট করার ফাইলগুলোকে তুলে নেয় '/dev/hda1' থেকে আর স্ল্যাকওয়ার তুলে নেয় '/dev/hda6' থেকে। এখানে পরপর তিনটে আলাদা আলাদা স্টেপ খেয়াল করুন। স্টেপ এক, যখন কাঁচা ডিস্ক থেকে তাকে ফাইল সিস্টেম বানানোর জায়গায় আনা হচ্ছে, পার্টিশন বানানো হচ্ছে, সেটা গোটা হার্ডডিস্ককে ব্যবহার করেও করা যেতে পারে, আবার তার একটা অংশকেও করা যেতে পারে, এই উদাহরণের মত। স্টেপ দুই, যেখানে, একটা বিশেষ অপারেটিং সিস্টেম সেই পার্টিশনে নিজের ফাইলসিস্টেম গড়ে তুলছে, এই ফাইলসিস্টেম মানে তার নিজের ডিরেক্টরি হায়েরার্কি এবং সেখানে ফাইলগুলোকে, এবং ফাইল রাখার বন্দোবস্তটাকে। এই কাজটা করা যেতে পারে নানা ধরনের ফাইলসিস্টেম ব্যবহার করে, মানে, ফাইল লেখার এবং রাখার স্বতন্ত্র প্রক্রিয়ায়, যেমন, রাইজারএফএস, এক্সএফএস, ইএক্সটিথ্রি, উইন্ডোজ-ফ্যাটথার্টিটু, ডস-ফ্যাটসিক্সটিন, ইত্যাদি। স্টেপ তিন, এক বা একাধিক পার্টিশনকে মিলিয়ে তৈরি হচ্ছে এক একটা আলাদা আলাদা অপারেটিং সিস্টেমের এক একটা আলাদা আলাদা ইউনিফায়েড ফাইল সিস্টেম। এতে কখনো, কোনো একটা ঐক্যবদ্ধ সিস্টেমের ক্রমানুসারী বা হায়েরার্কিকাল ফাইলব্যবস্থায় সবগুলো হার্ডডিস্কের সবগুলো পার্টিশনকেই নিয়ে আসা হতে পারে, যেমন আমার মেশিনের সুজে। আবার তা নাও হতে পারে, যেমন, স্ল্যাকওয়ারে, সুজের মূল পার্টিশনটাকে মাউন্ট করা হয়না, সুজে পার্টিশনটায় ফাইল বানানোর ব্যবস্থার নাম এক্সএফএস, যা কাজে লাগাতে গেলে অন্য একটা ড্রাইভার কারনেলে রাখার দরকার পড়ে, স্ল্যাকওয়ারে তা চাইলে করা যায়, আমি ইচ্ছে করেই করিনি, একবার দেখবার জন্যে কয়েকদিনের জন্যে করেছিলাম। কিন্তু এনভিডিয়া, যা আমার মাদারবোর্ড, ইন্টিগ্রেটেড ভিডিও কার্ড সাউন্ড কার্ড সহ, 'A7N266VM', তার ড্রাইভারে কিঞ্চিৎ বেদনা আছে, ঠিক ভাবে কারনেলে কম্পাইল হতে চায়না, অনেক অ্যাডজাস্ট করতে হয়, সুজে যেটা নিজেই করে দেয় — স্ল্যাকওয়ার একবার ব্যবহার করে দেখুন, ভারি মজা, প্রতিটি কনফিগারেশন আর অ্যাডজাস্টমেন্ট একদম নিজে হাতে ফাইল লিখে লিখে করতে হয়। আবার উইন্ডোজ যে ফাইলসিস্টেম দেখতে পায়, এবং ব্যবহার করে শুধু ফ্যাটথার্টিটু পার্টিশনদুটো, অন্য কোনো লিনাক্স পার্টিশন উইন্ডোজ ফাইলসিস্টেমে ঢোকেনা। একটা ইউনিফায়েড ফাইলসিস্টেমের সমগ্রে কোন কোন পার্টিশন কী ভাবে ঢুকবে, আমরা একটু বলেছি তাই নিয়ে, পরে মাউন্টের প্রসঙ্গে আরো আসবে — এটা নির্ভর

করে আমার ইচ্ছে এবং আমার অপারেটিং সিস্টেমের কারনেলের উপর, এইমাত্রই তো দেখলেন স্ল্যাকওয়ারের কারনেলের প্রসঙ্গে।

পার্টিশন কী বারবার তার উদাহরণ দেখেছি আমরা, এবার এর সংজ্ঞাটা নিয়ে একটু ভাবা যাক। একটা ভৌত হার্ডডিস্ককে যখন আমরা একাধিক লজিকাল টুকরোয় ভেঙে নিচ্ছি, তার এক একটাকে ডাকছি পার্টিশন বলে। লজিকাল বলতে যৌক্তিক এই অর্থে যে অপারেটিং সিস্টেমের একটা আস্ত অটুট হার্ডডিস্কের সঙ্গে কাজ করার যে যুক্তি বা লজিক সেটা এবার চালু থাকবে ওই পার্টিশনগুলোর উপরেও। মানে তাদের ভৌত একতাকে নিয়েই তারা এখন যৌক্তিক ভাবে অনেক হার্ডডিস্ক হয়ে উঠবে। একটা পার্টিশন তৈরি হয় কন্টিগুয়াস ব্লকদের নিয়ে। ব্লক মানে যে একক দিয়ে একটা হার্ডডিস্কের তথ্য রাখার সামর্থ্যকে মাপছি আমরা, সেই গোড়া থেকেই। এই ব্লকগুলোকে হতে হবে কন্টিগুয়াস বা একত্রে সম্বদ্ধ, মানে একই জায়গায়। একটা হার্ডডিস্কের এখানে কিছু ব্লক, ওখানে কিছু ব্লক, তাদের নিয়ে পার্টিশন তৈরি করা যায়না। এই একত্রে সম্বদ্ধ ব্লকগুলোকে মিলিয়ে তৈরি পার্টিশনটাকে অপারেটিং সিস্টেম একটা স্বতন্ত্র হার্ডডিস্ক হিশেবে গণ্য করে। হার্ডডিস্কের কোন অংশকে নিয়ে, কোন কোন ব্লক নিয়ে, একটা পার্টিশন তৈরি হচ্ছে, তার খুঁটিনাটিগুলো লেখা থাকে পার্টিশন টেবিলে।

আমরা একাধিক পার্টিশন করি, করতে হয় আমাদের, এর একটা বড় কারণ হল তথ্যের নিরাপত্তা, আগেই তো বলেছি, উইন্ডোজে থাকাকালীন, যখন ভাইরাসের ভয়ে কুঁকড়ে থাকতে হত সবসময়, কোনো বন্ধু কোনো মেল অ্যাটাচমেন্ট পাঠালেও খোলার আগে দিন দশেক ফেলে রাখতাম, যদি কোনো নতুন ভাইরাস থাকে, তাহলে তার ভাইরাস নিধক আপডেট বেরিয়ে যাওয়ার সময়টা দিতে, সবচেয়ে বেশি ভাইরাস তো আসে ওই অ্যাটাচমেন্টগুলো থেকেই, আরো যদি এমএসওয়ার্ড ফাইল অ্যাটাচমেন্ট হয়, তার মধ্যেকার মাইক্রোগুলোয় — ওইসময় তো বলেছি, একটা বাড়তি পার্টিশন রাখতেই হত, একটার অত্যাবশ্যক তথ্য অন্যটায় চালান করে দাও, তারপর গোটা ড্রাইভটা ফরম্যাট করে ফেলো। শুধু এই ভাইরাসভীতি নয়, ধরুন অন্য কোনো কেলোও যদি ঘটে একটা পার্টিশনে, হঠাৎ করে রাশিরাশি ব্যাড সেক্টর এসে সব ঘেঁটে গেল, তখন অন্য পার্টিশনটায় রাখা ফাইলগুলো বেঁচে যাবে। সেইজন্যে রেগুলার মূল কাজের পার্টিশনটার জরুরি ফাইলগুলোর ব্যাকআপ নিতে হয় অন্য আর একটা পার্টিশনে। আমার দুটো হার্ডডিস্কেরই জরুরি ফাইলগুলো একটা হার্ডডিস্কের থেকে অন্যটার কোনো পার্টিশনে ব্যাকআপ করার জন্যে একটা শেল-স্ক্রিপ্ট বানিয়ে রেখেছি, এক দিন দু দিন অন্তর অন্তর সেটা চালিয়ে দিয়ে চলে যাই, শেষ লাইন ‘poweroff’, ও নিজেই কাজ শেষ হলে মেশিন অফ করে দেয়।

পার্টিশন বাড়ানোর আর একটা কারণ হার্ডডিস্কের জায়গা আরো ভালো করে ব্যবহার করতে পারা। এটা খুব দরকার পড়ে যাদের অনেক ছোট ছোট সাইজের ফাইল ব্যবহার করতে হয়, যেমন বড় বড় নেটওয়ার্ক মেশিনগুলোয়, কোটি কোটি কুকি থাকে — বিভিন্ন ওয়েবসাইটের পরিচয়পত্র, সেইসব ফাইলগুলো খুব গুড়িগুড়ি সাইজের হয়। এবার তাদের জন্যে যদি ব্লক সাইজ স্বাভাবিক চার কিলোবাইটের মাপেই রাখা হয় তাহলে প্রতিটা ওই জীবানু সাইজের ফাইলের জন্যেই নষ্ট হতে থাকে চার কিলোবাইট করে জায়গা, কারণ, একটা ব্লকে একটার বেশি ফাইল রাখেনা সিস্টেম। এই জন্যে ওই ছোট ছোট ফাইল ব্যবহারের বিশেষ পার্টিশন বানানো যায় যাদের ব্লক সাইজ স্বাভাবিকের চেয়ে অনেক কম।

অনেকসময় পার্টিশন বানাতে হয় খুব বেশি কমপ্ল্যান বা হরলিকস খাওয়া ব্যবহারকারীদের শায়েস্তা রাখার জন্যেও। ধরুন হার্ডডিস্কের পাঁচ গিগাবাইট ভূমি আছে সেই পার্টিশনে যেখানে আমার সিস্টেমের ‘/home’ ডিরেক্টরি। আগেই তো বলেছি, গ্নু-লিনাক্সে আলাদা আলাদা ডিরেক্টরি আলাদা আলাদা পার্টিশনে মাউন্ট করার ব্যবস্থা করা যায়। আগেই দেখিয়েছি, ‘/home’ ডিরেক্টরির মধ্যে চারটে সাবডিরেক্টরি আছে, ‘/home/atithi’, ‘/home/dd’, ‘/home/manu’, আর ‘/home/piu’। এবার ধরুন ‘piu’ নামের ওই ইউজার ও ভুলভাল সব ফাইল এনে এনে, গানের কিম্বা সিনেমার বা অঙ্কের গাবদা গাবদা সব ডায়াগ্রাম, ভরেই চলেছে নিজের হোম মানে ‘/home/piu’ ডিরেক্টরিতে। ফাইলগুলো ভুলভাল, বুঝতেই পারছেন, ইউজার ‘dd’ আনলে যা হতনা। এবার, ‘/home/piu’ বেড়েই চলেছে। বাড়তে বাড়তে এমন একটা সময় চলে এল যে অন্য ইউজারদের আর সাধারণ ফাইলগুলো লেখার জায়গাটুকুও রইলনা। এধরনের বেআক্কেলে ব্যবহার সামলানোর একটা সহজ উপায় হল ‘/home/piu’ ডিরেক্টরির জন্যে একটা আলাদা পার্টিশন করে দেওয়া, ধরুন পাঁচ গিগাবাইটের চার ভাগের এক ভাগ মানে সোয়া এক জিবি সাইজের। এবং পষ্ট বলে দেওয়া, নে,

এবার কী করবি কর, তোর জায়গা মোট এত, যা রাখতে পারিস রাখ। এখনো করিনি এটা, আমার মত উদার লোক আর হয়না বলে।

একখানা আসমুদ্রহিমাচল পার্টিশনের জায়গায় বাংলা বিহার উড়িষ্যা টাইপের ছোট ছোট পার্টিশনে ফাইল অনেক কম টুকরো হয়, যাকে বলে ফ্র্যাগমেন্টেশন। ফ্রাগমেন্টেশন বাড়লে ডিস্ক থেকে ফাইল পড়ার বা ডিস্কে ফাইল লেখার সময় বেড়ে যায়, মানে সিস্টেম শ্লথ হয়ে পড়ে। এই সমস্যাটা অবশ্য ফ্যাটথার্টিটুতে যে ধরনের, গ্নু-লিনাক্সের ফাইলসিস্টেমগুলোয়, রাইজার বা এক্স বা ইএক্সটিথ্রি, আদৌ সেটা নয়, পরে দেখবেন। আর, একাধিক পার্টিশনের একটা জরুরত তো আগেই বলেছি, একাধিক অপারেটিং সিস্টেম রাখা। আমার সুজে স্ল্যাকওয়ার আর উইন্ডোজ এই তিনটে অপারেটিং সিস্টেমের জন্যে অন্তত তিন পিস পার্টিশন তো লাগবেই।

পার্টিশন এরকম বাড়িয়েই তোলা যায়, শুধু মাথায় রাখতে হয়, পার্টিশন বানানোর সময় একটার জন্যে ঘোষিত ভূমি যেন অন্যটার মধ্যে না ঢুকে যায়, পার্টিশন বানানো বদলানো বা নাড়াচাড়ার জন্যে যে সফটওয়ার প্যাকেজগুলো থাকে, যেমন 'fdisk', 'cfdisk', 'sfdisk', 'parted' ইত্যাদি, সেগুলোয় এটা হওয়ার কথাও নয়, তবে একাধিক সফটওয়ার দিয়ে আলাদা আলাদা সময়ে পার্টিশনগুলোর আকারআকৃতি বদলালে কখনো কখনো সমস্যা আসতে দেখেছি। এগুলো সবই ফ্রি সফটওয়ার, 'parted' তো গ্নু-র। 'cfdisk' তো আবার পিছনে 'fdisk'-কেই ব্যবহার করে, এনকার্সেস ব্যবহার করে ছবিতে দেখায় কনসোলেই। আমার দু একবার সমস্যা হয়েছে বাণিজ্যিক সফটওয়ার পার্টিশন-ম্যাজিক আর 'fdisk'-এর মধ্যে। তবে 'sfdisk' সত্যিই খুব জাঁদরেল, একটু বেশি ক্রিপ্টিক বা সাঙ্কেতিক যদিও, একটু বেশি ভালো করে ম্যানপেজ পড়ে নিতে হয়। আর অনেক ডিস্ট্রিবিউশনের ভিতরে নিজেদের অনেক প্যাকেজ দেওয়া থাকে যা দিয়ে পার্টিশনগুলোর সাইজ বাড়ানো কমানো যায়, কখনো কপিও করা যায়, তবে সেসব বেশি না-করাই ভালো। আপনি এই প্যাকেজগুলোর যেটাই ব্যবহার করুন আগে বার তিনেক গোটা ম্যানপেজ পড়ে নেবেন, নয়তো ঘেঁটে ফেলাটা অবশ্যম্ভাবী, আর পাকস্থলীতে র টাইটানিয়াম না-থাকলে ম্যানপেজ এক বারে হজম করা যায়না, এবং তথ্য হিশেবে বলে রাখি, একমাত্র আইনস্টাইন ছাড়া আর কারুর ভিসেরায়, এখনো, টাইটেনিয়াম পাওয়া যায়নি।

আমরা ম্যান আর ইনফো নিয়ে কথা বলেছি আগে। এখন আর একটু এগোনোর সময় এসেছে — এবার সিস্টেম হাউটু পড়ার সময় এসেছে, যে ডিস্ট্রিবিউশন আপনি ব্যবহার করছেন তাতে হাউটু-গুলো কোথায় আছে দেখে নিন। খোঁজার একটা সহজ কমান্ড আছে 'whereis', এছাড়া 'find' তো আছেই। 'find' ব্যবহারের অপশানগুলো আগে শিখে নিন ম্যানপেজ পড়ে। যেমন, যদি আপনি কমান্ড দেন, `find / -name howto` — এর মানে গোটা '/' ডিরেক্টরির সমস্ত সাবডিরেক্টরি বেয়ে সার্চটা ও করবে, কোথায় কোন ফাইল বা ডিরেক্টরির নামে 'howto' শব্দটা আছে। এখানে আপনি রেগুলার এক্সপ্রেশনও দিতে পারেন। ধরুন '*howto*' ইত্যাদি। এই 'find' কমান্ডটা কাজ করে বাস্তব ডিরেক্টরিতে নেমে নেমে বাস্তব ফাইলগুলোকে খুঁজে খুঁজে, তাই সময় একটু বেশি লাগে। 'find'-এর চেয়ে অনেক দ্রুত কাজ করে 'locate'। 'locate' বাস্তব ফাইল বা ডিরেক্টরিগুলো খোঁজে না, খোঁজে একটা ডেটাবেস। সেই ডেটাবেসটা আগে বানিয়ে নিতে হয় 'updatedb' কমান্ড দিয়ে। এই কমান্ডটা দিতে গেলে রুট হয়ে নিতে হয়। আপনি যদি অন্য কোনো ইউজার হয়ে ঢুকে থাকেন তাহলে 'su' কমান্ড দিন। তখন ও রুট পাসওয়ার্ড চাইবে, পাসওয়ার্ড দিয়ে দেওয়ার পর 'updatedb' কমান্ডটা কাজ করবে। ডেটাবেসটা তৈরি হতে কিন্তু বেশ একটু সময় লাগে। যদি অন্য কাজের তাড়া থাকে তাহলে কাজটা করুন ব্যাকগ্রাউন্ডে। `updatedb &` দিলে ও কাজটাকে পিছনে গোপনে গোপনে করতেই থাকবে, আপনাকে ডিস্টার্ব করবে না। 'Ctrl-F1' থেকে 'Ctrl-F6' টিপে অন্য কোনো ভৌতিক টার্মিনাল বা 'tty'-তে গিয়েও কাজটা করতে পারেন, ভৌতিক বা ভার্চুয়াল টার্মিনালের কথা মনে আছে? আর এই ডেটাবেস আপডেট করার কাজটা বেশ রসদকাঙ্খী, সিপিইউ-র উপর চাপ ফেলে, তাই ব্যাকগ্রাউন্ডে হওয়ার সময় অন্য কাজকে ডিস্টার্ব করতে পারে। যদি চান যে এই কাজটা ব্যাকগ্রাউন্ডে চলুক এবং আপনার অন্য কাজের সঙ্গে খুব ভালোমানুষের মত নাইসলি ব্যবহার করুক, তাদের কাজকে শ্লথ না-করে, না আটকে-রেখে, তাহলে কমান্ড দিন `nice -19 updatedb &`। এবার, ডেটাবেস তো তৈরি হয়ে গেল, হাউটু খুঁজবেন কী করে? কমান্ড দিন `locate howto`। সাবধান, আবার সেই রাশি রাশি লাইন হুড়হুড় করে নেমে যাচ্ছে পলক ঝপকতে। তার জন্যে কী করার আগেই তো বলেছি, এক হল লেস দিয়ে একটু একটু করে পড়া, বা রিডাইরেক্ট করে একটা ফাইল বানানো, সেটাকে পড়া, খুঁজে দেখা। খোঁজার কমান্ড 'grep' মনে আছে তো? এবার এই গোটা প্যারাগ্রাফে নতুন নতুন যে

জিনিষ এলো, চারটে এসেছে, এদের আপনি ম্যান বা ইনফো দিয়ে জেনে নিন। শুধু ওই '&' ব্যাপারটা পাবেন ব্যাশে, বা ব্যাশ-স্ক্রিপ্টিং নিয়ে হাউটুতে, যে হাউটু খোঁজার জন্যে আমরা ডেটাবেস আপডেট করলাম। কোনো কোনো ডিস্ট্রিবিউশনে সমস্ত লিনাক্স হাউটু একসাথে, যা টোটাল-হাউটু নামে পরিচিত, সেটা দেওয়া থাকে, যেমন সুজে বা ম্যানড্রেক। সুজেতে এর পাথটা হল '/usr/share/doc/howto/en/html/'। ম্যানড্রেকে যদ্দূর মনে হচ্ছে হাউটুর বানানটা ক্যাপিটাল কেসে। কোনো কোনোটায় প্যাকেজ ভিত্তিক হাউটু দেওয়া থাকে, মোটটা থাকেনা, সেইরকম অবস্থায় আপনি এটা নামিয়ে নিন 'www.tldp.org' বা 'www.linuxdocs.org' বা এফটিপি করে নামিয়ে নিন, 'anonymous' হয়ে, 'ftp://www.ibiblio.org/pub/Linux/docs/HOWTO' থেকে। 'tar.bz2' করা যে ফাইলটা সেটা সবচেয়ে সাইজে ছোট হয়। আমরা যারা ডায়ালআপ কানেকশনে কাজ করি তাদের পক্ষে সবচেয়ে ভালো। ফাইলনামের এক্সটেনশন 'tar.bz2' মানে কোঁকড়ানো সিন্দুক, এর থেকে ভেঙে মূল ফাইলগুলো বার করার কমান্ড হল 'tar xvjf', তবে তার আগে একটু . . . *খেজুরগাছে* . . . গেয়ে নেওয়া ভালো। এই হাউটুর মোট ভাণ্ডারটা রোজই বাড়ছে এবং বদলাচ্ছে, আমার কাছে শেষ যেটা নামানো আছে তাতে চারশো উনসত্তরটা হাউটু আছে। পিডিএফ বা ডকুমেন্ট ফাইলে বদলে নিলে যার বেশিরভাগেরই আকার 'A4' পঞ্চাশ পাতারও বেশি। আপনি বুঝতে পারছেন, সিস্টেম কত দূর অব্দি তার সকল নিয়ে বসে আছে সর্বত্যাগের আশায় — আপনাকে দিয়ে উজাড় হবে বলে। ভেবে নিন তাকে ভিখারি বানানোর মত আপনি যথেষ্ট গৌরী তো? এই হাউটুতেই আপনি পেয়ে যাবেন, এইমাত্র যে 'tar' আর 'bzip2'-র কথা বললাম, বা যে প্রসঙ্গে আমাদের আসা এই হাউটু-র কথায় — মানে হার্ডডিস্কের পার্টিশন নিয়ে নাড়াচাড়া, আপনি এই হার্ডডিস্ক আর পার্টিশন নিয়েই পাবেন চোদ্দখানা হাউটু। এর পরেও আপনার হার্ডডিস্ক এবং কারনেলের সম্পর্কের ডিটেইলস যদি জানতে চান তাহলে একটা ভারি ভাল উৎস হল আপনার সিস্টেমের ভিতরেই থাকা '/usr/src/linux/Documentation/'। এর মধ্যে দেখুন, আইডিই এবং স্কাসি সম্পর্কে দুটো আলাদা ডকুমেন্ট দেওয়া আছে। আর সামগ্রিক ভাবে ডিভাইস নিয়েও আছে। এবার ফেরত যাওয়া যাক পার্টিশনের কথায়।

১০.৩।। পার্টিশনের রকমফের

আগেই বলেছি আমরা একটা পার্টিশনে কেবল একরকমের ফাইলব্যবস্থা থাকতে পারে। সেই ফাইল লেখা পড়া ও রাখার সিস্টেম ইএক্সটিথ্রি হোক, রাইজারএফএস হোক, এক্সএফএস হোক, বা সোয়াপ পার্টিশনের নিজস্ব সোয়াপ ফাইল ব্যবস্থা হোক। বা গ্নু-লিনাক্স-এর বাইরের যেসব ফাইল ব্যবস্থার পার্টিশনকে মাউন্ট করে নেওয়া যায় কোনো ডিরেক্টরিতে, সেই মাইক্রোসফটের ফ্যাটথার্টিটু বা এনটিএফএস বা সান মাইক্রোসিস্টেমের ইউএফএস হোক, বা এরকম আর কিছু। প্রত্যেক ধরনের পার্টিশনের নিজের একটা কোড থাকে। যেমন ইএক্সটিটু পার্টিশনের কোড '0x83', সোয়াপ পার্টিশনের কোড '0x82', ইত্যাদি। কোন পার্টিশনের কী কোড এটা জানার সহজ উপায় হল 'fdisk'-এ ঢুকে 'l' মারা মানে পার্টিশন কোডের লিস্ট দেখাতে বলা।

হার্ডডিস্ককে কটা পার্টিশনে ভাঙা যাবে, ইন্টেল বা ইন্টেল ধাঁচে নির্মিত মেশিনের বেলায়, এর একটা সীমা গোড়া থেকেই ছিল, মূল পার্টিশন টেবিলটা লেখা হত হার্ডডিস্কের বুট সেক্টরে এবং তাতে জায়গা ছিল মাত্র চারটে পার্টিশনের নাম লেখার। এই ধরনের প্রাথমিক পার্টিশনকে বলে প্রাইমারি পার্টিশন। এদের সংখ্যা হতে পারে সর্বোচ্চ চার। কিন্তু স্বাভাবিকভাবেই এই কটা পার্টিশনে কুলোনো সম্ভব না-হলে তখন পার্টিশন বাড়াব কী করে? তার জন্যে এল লজিকাল বা যৌক্তিক পার্টিশন। যেখানে একটা ওই চারটে প্রাথমিক ভৌত প্রাইমারি পার্টিশনের কোনো একটাকে আবার ভেঙে ফেলা হচ্ছে কতকগুলো ছোটতর পার্টিশনে। তাদের আমরা ডাকছি লজিকাল পার্টিশন বলে। এতে করে ওই চার পার্টিশনের সীমাটাকে ভাঙা গেল। যে প্রাইমারি পার্টিশনের মধ্যে এবার আমি ছানা পার্টিশনগুলোকে গজাচ্ছি, তাকে এখন ডাকব এক্সটেন্ডেড বা পরিবর্দ্ধিত পার্টিশন বলে। এর আলাদা করে কোনো ফাইল ব্যবস্থা থাকেনা, কোনো ফাইলও, এই এক্সটেন্ডেড পার্টিশন ধরে রাখে এর ভিতরের লজিকাল পার্টিশনগুলোকে। এই ধরনের পার্টিশনের কোড হল '0x05'। এখানে লজিকাল পার্টিশন আর প্রাইমারি পার্টিশনের আর একটা তফাত বলে রাখা ভাল। মূল হার্ডডিস্কে কোনো একটা নির্দিষ্ট পরিমাণ ভূমিকে আমি চাইলে ছেড়েও রাখতে পারি, কোনো পার্টিশন না বানিয়ে, পরে বানাব প্রয়োজনমত। তাই প্রাইমারি পার্টিশনগুলোকে যে কন্টিগুয়াস বা পরপর সংলগ্ন হতেই হবে এমন কোনো বাধ্যতা নেই। কিন্তু একটা এক্সটেন্ডেড পার্টিশনের মধ্যে লজিকাল পার্টিশনগুলোকে ঠিক পরপর বা কন্টিগুয়াস হতেই হবে। প্রত্যেকটা লজিকাল পার্টিশনের মধ্যেই পরবর্তী লজিকাল পার্টিশনের একটা পয়েন্টার বা চিহ্নক থাকে। এই করে

পরপর অনন্ত সংখ্যক লজিকাল পার্টিশন থাকতে পারে। কিন্তু গ্নু-লিনাক্সের নিজস্ব কিছু নিষেধ আছে এখানে, একটা স্কাসি হার্ডডিস্কে গ্নু-লিনাক্সে মোট পার্টিশন থাকতে পারে পনেরোটা অব্দি, আর আইডিই হার্ডডিস্কে থাকতে পারে তেষট্টিটা অব্দি।

প্রাথমিক ফাইলসিস্টেম পার্টিশনগুলো ছাড়া অন্য যে পার্টিশন থাকে সেটা হল সোয়াপ পার্টিশন, সোয়াপ ফাইল লেখার জন্যেই তুলে রাখা জমিজিরেত, এক কথায় সোয়াপস্পেস বা পরিবর্তভূমি। এক আর দুই নম্বর দিনের আলোচনাগুলো মনে করুন — কম্পিউটারে যখনি আমি কোনো প্রোগ্রাম চালাচ্ছি, একটা প্রসেস চালু হচ্ছে, আর এই প্রসেসের জন্যে কম্পিউটারের সিপিইউ ধার্য করে দিচ্ছে র‍্যাম ব্লকের একটা নম্বর, ঠিক যেভাবে আমরা দেখিয়েছি দুই নম্বর দিনে, যে ব্লক গুলোকে ডাকা হয় পেজ বা স্মৃতির পাতা বলে, আগেই বলেছি। এই পাতায় যে কোডটা তুলে ফেলা হয়, সেটা খুব আশু ভবিষ্যতেই কাজে লাগবে সিপিইউ-র। সিপিইউ-র ভিতরকার রেজিস্টারগুলোকে কাজে লাগিয়ে সিপিইউ সেখানে বারবার পড়বে আর লিখবে, এক নম্বর দিনের আলোচনায় দেখিয়েছি আমরা। এই কোডগুলোকে ডাকা হয় 'ওয়ার্কিং সেট' নামে, সত্যিই তো ওগুলো ভারি কাজের জিনিষ। গ্নু-লিনাক্স ধরে নেয় যে সদ্য কাজে লাগানো জিনিষগুলো আবার শিগ্গিরি কাজে লাগবে, তাই তাদের র‍্যাম মেমরির কোড-পেজে তুলে ফেলে। কিন্তু যদি একইসঙ্গে অনেকগুলো প্রসেস চলতে থাকে — মাল্টিপ্লেক্সিং-এর কথা মনে আছে? অনেকগুলো প্রসেস একত্রে চলাকালীন কারনেল চেষ্টা করে র‍্যামকে যথাসাধ্য মুক্ত করে তুলতে, যাতে পরপর একের পর এক কাজকে সে র‍্যামে নিয়ে আসতে পারে। এই র‍্যামের ভার লাঘব করার জন্যেই থাকে সোয়াপ ফাইল, আগেই বলেছি। কাজের দিক থেকে বলতে গেলে, র‍্যামকে আসলে বাড়িয়ে তোলে সোয়াপ স্পেস বা পরিবর্ত ভূমি, র‍্যামের পরিবর্তে এই ভূমিতেই অস্থায়ী ভাবে লেখা হয় র‍্যামের ওই কোডপেজগুলো। যদিও এখানে সবচেয়ে বড় গ্যাঁড়াকলটা হল আইও, ইনপুট/আউটপুট। এই 'আইও' লিখতে গেলেই আমার মনে হয় আমাদের জিএলটির সেই সভ্যের কথা যে মুহুর্মুহু 'আইও আইও' করে, কিন্তু এই লেখাটার সঙ্গে তার বিন্দুমাত্র সম্পর্কও নেই, কারণ বাংলাই জানেনা, দাঁড়ান, ওর পুরো নামটা এখানে লিখে দি, যাতে পড়তে গিয়ে আপনাকে অন্তত একবার 'আইও' বলে উঠতেই হয় — পোনথিরানাভকারাসু বিশালাকষ্মি মুরুগেশন, আমরা ডাকি আরাসু বলে। যাইহোক এই ইনপুট আউটপুটের শ্লথতাই এই র‍্যামপেজ সোয়াপফাইলে বারংবার প্রয়োজনমত লিখে ফেলার এবং পরে যখনই কাজ পড়বে পড়ে ফেলার ব্যাপারে একমাত্র বাধা। র‍্যামে কোনো কিছু লেখা যে অবিশ্বাস্য দ্রুততায় করতে পারে সিপিইউ সেটা কখনোই হার্ডডিস্কে সোয়াপ ফাইলের বেলায় সম্ভব নয়, তাই সোয়াপ ফাইল র‍্যামকে বাড়িয়ে তোলে বটে, কিন্তু র‍্যামের তুলনায় বড্ড বেশি ধীরে কাজ করে। অনেক প্রসেসের ভিতর অনেক ভাগে ভাগ হতে হতে মেমরি যখন লম্ফর টিমটিমে হলুদ আগুনের মত কোনোক্রমে মরে না গিয়ে বেঁচে থাকার মত পরিমাণের হয়ে পড়ে, তখন কারনেলকে বাধ্য হয়ে একটা প্রসেসের কাজের জিনিষ মানে ওয়ার্কিং সেটের কোড-পাতাগুলো সরিয়ে ফেলতে হয়, আর একটা প্রসেসের ওয়ার্কিং সেট মেমরি পাতায় তোলার আগে। সরিয়ে নিয়ে রাখে ওই সোয়াপ ফাইলে। একে বলে থ্র্যাশিং, মানে এক অর্থে কেলিয়ে বার করে দেওয়া। কিন্তু আগেই তো বলেছি বারবার, সিপিইউ আর র‍্যামের হটলাইন বেয়ে তথ্য যে স্পিডে লেখা-পড়া হয়, তার ধারেকাছেও আসেনা সোয়াপফাইলের হার্ডডিস্কের আইও। তাই, যতই সোয়াপফাইল র‍্যামকে বাড়িয়ে তুলুক, পর্যাপ্ত র‍্যাম ইজ পর্যাপ্ত র‍্যাম, তার জায়গা কদাচ সোয়াপ ফাইল নিতে পারেনা। গ্নু-লিনাক্সে সচরাচর র‍্যাম আর সোয়াপ ফাইল যোগ করে ভারচুয়াল বা ভৌতিক মেমরির আকার ধরা হয়। তার মানে ধরুন যদি দুশোছাপ্পান্ন মেগাবাইট র‍্যাম হয়, আর পাঁচশো চল্লিশ মেগাবাইট সোয়াপস্পেস রাখা হয়, তাহলে ভৌতিক স্মৃতির আকার দাঁড়াচ্ছে সাতশো ছিয়ানব্বই এমবি। সচরাচর ধরে নেওয়া হয় যে র‍্যাম যা হবে তার দ্বিগুণ হওয়া উচিত সোয়াপ স্পেস। র‍্যাম খুব বেশি বা খুব কম হলে এই হিশেবটা নাটকীয়ভাবে বদলে যায়। তবে এই হিশেবটারও আভ্যন্তরীন যুক্তি কী আমি জানিনা, তবে চলে কথাটা, এবং আমিও মোটামুটি মেনে চলি।

এবার প্রশ্ন ওঠে ঠিক কী কী রকমের কটা পার্টিশন আমার লাগবে। আমরা আগেই যা বলেছি, মাস্টার বুট রেকর্ড বা এমবিআরটা লেখা থাকে হার্ডডিস্কের একদম প্রথমতম ভূমিটায়। এই মাস্টার বুট রেকর্ডকে লজিকাল পার্টিশনে লেখা যায়না, তাই যদি আমরা ওই হার্ডডিস্ক থেকে একটাও অপারেটিং সিস্টেম বুট করতে চাই, তাহলে অন্তত একটা প্রাইমারি পার্টিশন লাগবেই। ওই হার্ডডিস্ক থেকে বুট করতে চাইলে শুধু একটা প্রাইমারি পার্টিশন থাকতে হবে তাই নয়, আরো থাকতে হবে একটা বা একাধিক সোয়াপ পার্টিশন। এবং এর সঙ্গে ইচ্ছে হলে প্রয়োজন হলে লজিকাল

পার্টিশন থাকতেই পারে, এর কোনো শর্ত নেই। আর, সেই ড্রাইভ থেকে যদি বুট করা না হয় তাহলে সেটা গোটাটা একটা পার্টিশন না একাধিক পার্টিশন, পার্টিশনটা প্রাইমারি না লজিকাল না সোয়াপ তার কোনোটারই কোনো শর্ত নেই। এর পরেও কিছু কথা আছে যেগুলো সেই অর্থে বাধ্যতামূলক না হলেও মেনে চলাটাই প্রথা। যেমন ধরুন আপনার '/boot' ডিরেক্টরিটা যে পার্টিশনে মাউন্ট হয়, মানে আপনার বুট পার্টিশন, যেখান থেকে সিস্টেম বুট করার সময় তার কারনেল এবং কারনেল সংক্রান্ত ফাইল পত্তর পড়ে, সেই পার্টিশনটাও প্রাইমারি পার্টিশন করাটাই প্রথা, এতে কোনো মেজর ঘাপলা ঘটলে পরিস্থিতিটা সামলানো সহজ হয়। আমি নিজেও এটা সবজায়গায় মানিনি, অন্তত শখানেক জায়গায়, নিজেরটা ছাড়াও ইনস্টলেশন করেছি তো বটেই, এদের অনেকগুলোতেই অনেক বিচিত্র কনফিগারেশন বহুসময় বাধ্যতামূলক ভাবে, করতে হয়েছে, সেসব বললে বিশুদ্ধতাবাদী লিনাক্সীরা বাড়ি এসে আমায় কেলিয়ে যাবে। তারা তো প্রায় সবসময়ই বলে এই পার্টিশনটাকে ইক্সটিটু পার্টিশন করতে, কিন্তু আমার নিজের সিস্টেমে তো গোড়া থেকেই, যখন ম্যানড্রেক করতাম, এখন সুজেতেও, বুট পার্টিশনটা সবসময়েই এক্সএফএস, যা সবচেয়ে বিপজ্জনক বলে পরিচিত। এখনো কোনো মেজর কেলো ঘটেনি। তবে, যারা বহুব্যবহারকারীর মেইনফ্রেম সিস্টেম চালায়, বহু ধরনের জটিলতা, তাদের তো মাইনর কেলো বলেই কিছু নেই, কেলো মানেই মেজর। তারা এসব ছ্যাবলামি করার সাহস পায়না। আমার আর কী, অন্য হার্ডডিস্কে ব্যাকআপ আছে, ঘেঁটে গেছে তো ভি আচ্ছা, রিইনস্টল করে নাও কাকা। আর গোটা '/home' ডিরেক্টরিটাই টার-বিজেডটু করা থাকে। জাস্ট আনটার করো, এবং ফের হুবহু আগের মত। তারপরে আর একটা প্রথা আছে, সেটা লিলোর। অন্য কোনো বুটলোডার যদি কেউ ব্যবহার করে, যেমন গ্রাব, বা কোনো থার্ড পার্টি প্যাকেজ, তাহলে সেটা প্রযোজ্য নয়। লিলো দিয়ে বুট করলে বুট পার্টিশনটা প্রথম ১০২৪ সিলিন্ডারের মধ্যে রাখাই প্রথা। অনেকসময়ই তা হয়না, আলতো একটু বকে দেয় সিস্টেম, কোনো গন্ডগোল ঘটতে তো দেখিনি তেমন। ভালো না, এসব ভালো না, বলেছেন শাক্যমুনি, চলো ভালো হই, জমকালো হই, বড়দার বাতেলা শুনি।

এবার দেখুন তো, আমার মেশিনের হার্ডডিস্কের পার্টিশনগুলোর খুঁটিনাটি, যেটা পাওয়া যায়, '`fdisk -l`' করে, যে তালিকাটাকে বেশ অনেকটা ছাঁটকাট করে ছয় নম্বর দিনের ছয় নম্বর সেকশনে দিয়েছিলাম, এখন সেটা পুরোটা পড়ে বোঝার মত আলোচনা আমাদের হয়েছে, দেখুন তো আপনি বুঝতে পারেন কিনা। ওইখানে দেওয়া তালিকাটায় আমি সরাসরি এক্সটেনডেড পার্টিশনের লাইনদুটো উড়িয়ে দিয়েছিলাম, আর ডিস্ক, সিলিন্ডার, ব্লক আর ট্র্যাকের খুঁটিনাটিগুলো দিইনি। শুরুর সিলিন্ডার নম্বর, শেষের সিলিন্ডার নম্বর দিইনি, ব্লকের সংখ্যা দিইনি। এবং বোঝার সুবিধের জন্যে জুড়ে দিয়েছিলাম মাউন্টপয়েন্টগুলো। দেখুন তো এবার গোটাটা বুঝতে পারেন কিনা? একটু আধটু অসুবিধায় একটু খেজুরগাছে চড়তে পারেন।

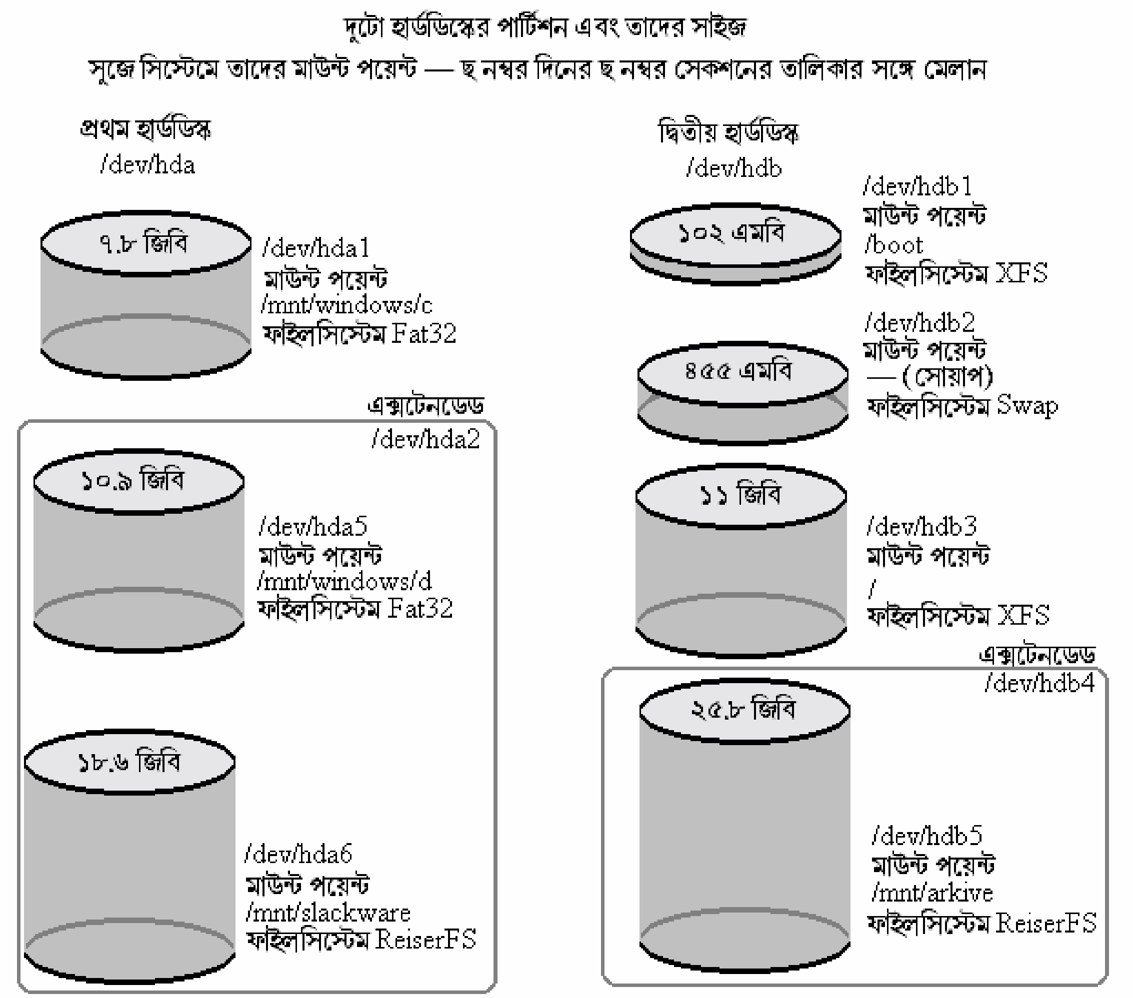
```
Disk /dev/hda: 40.0 GB, 40060403712 bytes
255 heads, 63 sectors/track, 4870 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot    Start       End    Blocks   Id  System
/dev/hda1   *         1      1020   8193118+   b  Win95 FAT32
/dev/hda2          1021      4870  30925125    f  Win95 Ext'd (LBA)
/dev/hda5          1021      2448  11470378+   b  Win95 FAT32
/dev/hda6          2449      4870  19454652   83  Linux

Disk /dev/hdb: 40.0 GB, 40060403712 bytes
255 heads, 63 sectors/track, 4870 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot    Start       End    Blocks   Id  System
/dev/hdb1   *         1        13    104391   83  Linux
/dev/hdb2            14        71    465885   82  Linux swap
/dev/hdb3            72      1508  11542702+  83  Linux
/dev/hdb4          1509      4870  27005265    5  Extended
/dev/hdb5          1509      4870  27005170   83  Linux
```

এই গোটা তালিকাটাকে মিলিয়ে মিলিয়ে পড়ুন, আগের দিনের মানে ছ নম্বর দিনের ছ নম্বর সেকশনে দেওয়া পার্টিশনের তালিকাটার সঙ্গে। দেখুন তো দুটো টেবিলকে মেলাতে পারছেন কিনা? শুধু একটা জিনিষ বলুন তো, ছ নম্বর দিনের ছ নম্বর সেকশনের ওই তালিকায়, '/dev/hda1'-এর পরেই কেন '/dev/had5', আর '/dev/hdb3'-এর পরেই কেন '/dev/hdb5', সেটা কি আপনি বুঝতে পেরেছেন? না-পেরে থাকলে আগের সেকশনটা আরো একবার পড়ুন। এই প্রশ্নটা প্রথম প্রথম গ্নু-লিনাক্সে আসা অনেকেই করে।

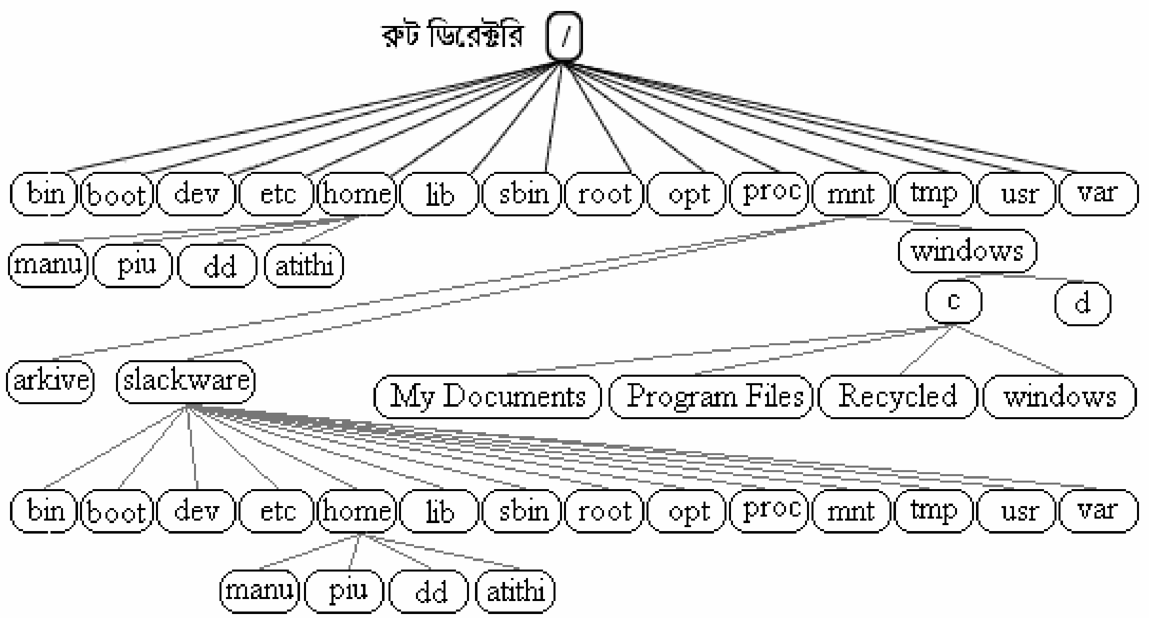


এবার আমার হার্ডডিস্কের পার্টিশনগুলোর একটা ছক ছবিতে দিই, দেখুন তাতে আপনার বুঝতে একটু সুবিধে হয় কিনা। একটু আগের তালিকাটা আর আগের দিনের তালিকাটাকে মিলিয়ে ছবিটা দেখুন। ছবিতে কিন্তু আমরা পরিমাপযোগ্যতা বা স্কেল রাখিনি, নিজে এঁকে দেখান তো একই সাথে পাশাপাশি একটা ১০২ এমবি আর একটা ২৫.৮ জিবি পার্টিশন, মানে একটা অন্যটার ২৫৯ গুণ?

১১।। ফাইলসিস্টেম

পার্টিশন তো হল, এবার? সেই পার্টিশনভূমি তো এখন শূন্য। এবার সেখানে রাজ্য জেলা মহকুমা পাড়া কায়েম করতে হবে, চোর এবং নেতা বানাতে হবে, মস্তান এবং পুলিশ, কার আদেশ কোথা থেকে কোথায় যায়, কোন রুটে, সেই গতিপথগুলো বানাতে হবে, ফাইলসিস্টেম। আর এই যে কোনো কিছু তাদের উপর করে দেওয়ার জন্যে, বারবার শূন্য করে ভরে দেওয়ার জন্যে, নির্বাক জনআধিক্যের ভার, শ্যাডো অফ দি সাইলেন্ট মেজরিটিজ, সেই আড্ডিক্মেমেরি বাইটকূল তো আছেই, আসমুদ্রহিমাচল জনগনমনের মত।

প্রত্যেক পার্টিশনেই একটা করে ফাইলসিস্টেম তৈরি করতে হবে। আমরা আগেই বলেছি একটা গ্নু-লিনাক্স ফাইলসিস্টেমের মধ্যে একাধিক ফাইলসিস্টেম ঢুকে আসতে পারে, এসে থাকে। যাদের প্রত্যেকের আবার নিজস্ব কাঠামো, নিজস্ব শিকড়, নিজস্ব ডালপালা থাকে, সবসময়ে এটা আলাদা করে খেয়াল থাকেনা। কারণ, তাদের সবাইকে মিলিয়ে একটা ঐক্যবদ্ধ ফাইলসিস্টেম তৈরি করা হয়, তার এক একটা অংশে এরা বিরাজ করে। যেমন, আমার সিস্টেমে '/mnt' ডিরেক্টরির কথা বলেছি, সেখানে '/mnt/arkive', '/mnt/slackware', '/mnt/windows' এই তিনটে সাবডিরেক্টরি আছে। ছ নম্বর দিনের আমাদের পুরোনো পরিচিত ছবিটাকেই একটু বদলানো চেহারায় আর একবার দেখা যাক, এবার শুধু ইউজার 'dd'-র হোম ডিরেক্টরি '/home/dd'-র ভিতরকার ডিটেইলসটা ছেঁটে দিয়েছি, আর তার জায়গায় '/mnt/slackware' এবং '/mnt/windows' ডিরেক্টরির ডিটেইলসটা একটু এনেছি, জাস্ট এক স্টেপ নিচে অব্দি।

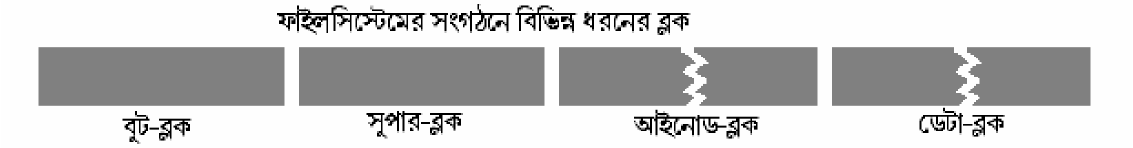


মূল ঐক্যবদ্ধ ফাইলসিস্টেমের '/' এখানে সুজের। এখানে দেখুন, '/mnt/slackware' ডিরেক্টরির ভিতরে সাবডিরেক্টরি কাঠামোটা প্রায় মূল '/' ডিরেক্টরির ভিতরকার কাঠামোর সঙ্গে এক। কিছু তফাত আছে, সুজে আর স্ল্যাকওয়ারের, কিন্তু জটিলতা এড়াতে আমি সেটা বাদ দিয়েছি। স্ল্যাকওয়ারের '/home' ডিরেক্টরির ভিতর আবার সেই একই চারজন ব্যবহারকারীর হোম ডিরেক্টরি। ছ নম্বর দিনের তালিকাটা দেখুন, সুজে সিস্টেমে '/mnt/slackware' ডিরেক্টরিটা মাউন্ট হয় '/dev/hda6' পার্টিশনে। এই '/dev/hda6' পার্টিশনটাই আবার '/' ডিরেক্টরি হয়ে ওঠে যখন স্ল্যাকওয়ার সিস্টেমে বুট করা হয়, এই গোটা ব্যাপারটা আপনি ভালো করে বুঝতে পারবেন মাউন্ট নিয়ে আমাদের আলোচনাটা হয়ে গেলেই। এখন একটু দেখে রাখুন। যারা উইন্ডোজ সিস্টেমের সঙ্গে পরিচিত তারা এখানে '/mnt/windows' ডিরেক্টরির ভিতর সাবডিরেক্টরিগুলোকেও চিনতে পারবেন। '/mnt/windows/c' ডিরেক্টরিতে মাউন্ট হয় '/dev/hda1' পার্টিশনটা, ছ নম্বর দিনের তালিকাটা থেকে মিলিয়ে নিন। '/mnt/windows/d' ডিরেক্টরিতে মাউন্ট হয় '/dev/hda5', আর '/mnt/arkive' ডিরেক্টরিতে মাউন্ট হয় '/dev/hdb5'। একটু বাদেই আমরা এই মাউন্ট প্রসঙ্গে আসছি।

অর্থাৎ, সুজের গোটা গ্নু-লিনাক্স ফাইলসিস্টেমের সংগঠনটার একটা নিজস্ব চরিত্র আছে। সেই চরিত্রটা এবার আমরা বোঝার চেষ্টা করব, আমরা শুরু করব ওই ব্লক থেকে, আগেই বারবার বলেছি, কম্পিউটারের তথ্যকে বোঝার এবং নাড়াচাড়ার একটা একক। একটা ব্লক মানে ১০২৪ বাইট। একবার হিশেব করে নিন একটু আগের তালিকার থেকে, শুধু খেয়াল রাখবেন যে, এক্সটেন্ডেড পার্টিশনটা এক বা একাধিক লজিকাল পার্টিশনে ভাঙা হয়ে থাকলে, এক্সটেন্ডেড পার্টিশনের মাপটা অন্যগুলোর মধ্যেই রয়েছে, মোট যোগ করে দিলে, এক্সটেন্ডেড পার্টিশনের মাপটা কিন্তু দুবার করে চলে আসে। আর ইউনিট বলে যেটা উল্লেখ করা হয়েছে সেটা হল একটা সিলিন্ডারের মোট বাইট ধারণ করার সামর্থ্য। এই সামর্থ্যটা দিয়ে মোট হার্ডডিস্কের বাইটসাইজকে ভাগ করে দেখুন, পাবেন চার হাজার আটশো সত্তর —

যেটা হল এই হার্ডডিস্কের সিলিন্ডারের সংখ্যা। হার্ডডিস্কের প্রতিটি পার্টিশনকেই দেখুন, একদিকে দেওয়া হয়েছে সিলিন্ডারের নিরিখে, কত নম্বর সিলিন্ডার থেকে শুরু, কত নম্বর সিলিন্ডারে শেষ। আর প্রতিটি পার্টিশনের তথ্য ধারণ করার সামর্থ্যটা দেওয়া আছে ব্লকের হিশেবে। এই ব্লকের ভিত্তিতেই তৈরি হয় গ্নু-লিনাক্স ফাইল সিস্টেমের সংগঠন। এই ব্লককে আবার ভাগ করা হয় চারটে আলাদা আলাদা রকমে।

এক, বুট ব্লক। বুট ব্লক বলতে বোঝায় যেখানে একটা ছোট বুট প্রোগ্রাম থাকবে এবং পার্টিশন টেবিলটা লেখা থাকবে। দুই, সুপারব্লক। সুপারব্লকে থাকে ফাইলসিস্টেমটার বিষয়ে সামগ্রিক তথ্য। এখানে একটা ফাঁকা আইনোডের তালিকা রাখা হয়, যেসব আইনোড ব্যবহৃত হয়নি, কারনেল চাইলেই ফাইল লেখার জন্যে ব্যবহার করা হতে পারে। আর রাখা হয় সেইসব ডেটা বা তথ্য ব্লকের তালিকা যারা ফাঁকা আছে, যেখানে কারনেল চাইলেই ফাইল বানাতে পারে, তথ্য লিখতে পারে। তিন, আইনোড ব্লক। আইনোড ব্লকের এলাকায় থাকে ফাইলসিস্টেমের প্রতিটি ফাইলের জন্যে একটা তালিকা বা টেবিল। সেখানে ফাইলের এবং ডিরেক্টরির প্রতিটি তথ্য লেখা থাকে, শুধু ফাইল বা ডিরেক্টরির নামটা বাদ দিয়ে। চার, ডেটা ব্লক। ডেটা বা তথ্য ব্লকে থাকে যাবতীয় ফাইল। সে ফাইল অপারেটিং সিস্টেমের বানানোই হোক, বা কোনো প্রোগ্রামের, বা কোনো ইউজারের। সে ফাইল রেগুলার ডেটা ফাইলই হোক, বা প্রোগ্রাম ফাইল। সমস্ত ধরনের ফাইলেরই বাসস্থান এই ডেটা ব্লক। এখানে ছবিতে আমরা আইনোড-ব্লক আর ডেটা-ব্লককে



ভেঙে দেখিয়েছি এটা বোঝাতে যে এরা অনির্দিষ্ট দৈর্ঘের হতে পারে, এবং স্বাভাবিকভাবেই আমরা আশা করতে পারি যে বুট-ব্লক বা সুপার-ব্লকের চেয়ে পরিমাণে এরা অনেক বেশি হবে।

১১.১।। বুট ব্লক এবং সুপার-ব্লক

একটা পার্টিশনের ফাইলসিস্টেমের একদম গোড়াতেই থাকে বুট-ব্লক। এর পরেই আসে সুপার-ব্লক। পাঁচ নম্বর দিনে বুট প্রসেসের আলোচনায়, এবং পরেও আমরা যে এমবিআর বা মাস্টার-বুট-রেকর্ডের কথা এনেছি, এই বুট ব্লকই তার বাসস্থান। বুট ব্লকে লেখা থাকে একটা পার্টিশন টেবিল, এবং বুটিং প্রোগ্রামের হালহদিশ। সেই বিষয়গুলো তো মোটামুটি আমাদের পরিচিত, বুট ইনিট কারনেলের সূত্রে। ইচ্ছে হলে একবার পাঁচ নম্বর দিনের আলোচনাটা একটু উল্টে আসতে পারেন। একটা সিস্টেম যখন বুট করে, সিস্টেমের বায়োস রম চিপে লেখা আদেশ মোতাবেক বায়োস প্রথমেই জরিপ করে নেয় প্রথম হার্ডডিস্কের অবস্থাটা, এবং বুট-ব্লকে লেখা গোটাটা সিস্টেমের মেমরিতে তুলে নেয়। তারপর নিয়ন্ত্রণ ছেড়ে দেয় বুটস্ট্র্যাপিং প্রোগ্রামের হাতে, আমরা যেটা লিলো দিয়ে দেখিয়েছি, এছাড়া গ্রাব হয়, অন্য থার্ডপার্টি সফটওয়ারও হয়। লিলো বা অন্য সেই বুটস্ট্র্যাপিং প্রোগ্রাম এবার কারনেল লোড করা শুরু করে, মনে করার চেষ্টা করুন, '/boot' ডিরেক্টরি থেকে '/boot/vmlinuz'। মনে করতে পারছেন, যার কণ্ঠস্বর শুনেছিলেন আপনি ক্যাট করে। কাজে যতই দড় হোক, গলা হিশেবে অত্যন্ত থার্ড ক্লাস, না-বলে উপায় নেই। একটা কথা — এই বুটলোডারটা থাকে কিন্তু রুট ফাইল সিস্টেমের গোড়ায়। অন্য ফাইলসিস্টেমগুলোর বেলায় এই বুট-ব্লকটা থাকে শূন্য। যেমন, আমাদের এই সিস্টেমে এমবিআর লেখা হয়েছে প্রথম হার্ডডিস্কের প্রথম পার্টিশনের গোড়ায়, আমরা এখন জানি, তার নাম, '/dev/hda1', অন্য পার্টিশন গুলোর বুট-ব্লক আছে শূন্য। এই কথাটা আমরা পাঁচ নম্বর দিনের আলোচনায় বলে আসতে পারিনি। এর মধ্যেই আমরা একটা গ্নু-লিনাক্স সিস্টেমকে অনেকটাই বুঝতে পারছি, দেখেছেন?

বুট-ব্লকের পরে এবং আইনোড ব্লকের আগে, একটা পার্টিশনের একটা ফাইলসিস্টেমে আসে সুপারব্লক। গ্নু-লিনাক্স অপারেটিং সিস্টেমের পার্টিশন পিছু হিশেব-নিকেশের খাতা বলতে পারেন। পার্টিশনটার তথ্যধারণের সামর্থ্য কতটা, তার কতটা ব্যবহৃত হয়েছে, কতটা জায়গা এখনো ব্যবহারযোগ্য আছে, কতগুলো আইনোড-ব্লক এবং ডেটা-ব্লক আছে যাতে এখনো ফাইল বানানো যায় — এই সমস্ত লিখে রাখার খাতা হল সুপারব্লক। এখানে যদি লেখায় ভুল হয় গোটা অপারেটিং সিস্টেমটাই ভুলভাল চলবে, মানে চলবেই না। যে যে বিষয়গুলো লিপিবদ্ধ থাকে সুপারব্লকে তার মধ্যে আছে — এক, এই পার্টিশনের ফাইলসিস্টেমের সাইজ। দুই, ফাইলসিস্টেমে মোট লজিকাল ব্লক কতটা আছে। এই লজিকাল ব্লকের ব্যাপারটা, আগেও এসেছে, আমরা এর পরেই আইনোড এবং ডেটা-ব্লকের আলোচনায় বুঝতে

পারব। তিন, শেষ কবে ফাইলটা বদলানো বা আপডেট করা হয়েছিল। চার, এই পার্টিশনে মোট কতগুলো ডেটা-ব্লক এখনো ফাঁকা আছে, এবং এখুনি তথ্য লেখা যায় এমন ডেটা-ব্লকের একটা আংশিক তালিকা। পাঁচ, এই পার্টিশনে মোট কতগুলো আইনোড এখনো ফাঁকা আছে, এবং এখুনি ব্যবহার করা যায় এমন আইনোড-ব্লকের একটা আংশিক তালিকা। ছয়, এই পার্টিশনের ফাইল সিস্টেমটা কী অবস্থায় আছে সেই সংক্রান্ত তথ্য, মানে সব ঠিকঠাক আছে, না কিছু জায়গা ঘেঁটে আছে।

একটা পার্টিশনের একটা ফাইলসিস্টেমকে মোট ঐক্যবদ্ধ ফাইলসিস্টেমের আভ্যন্তরীন করার সময় কারনেল এই সুপারব্লকের তথ্যটাকে সিস্টেমের মেমরিতে তুলে নেয়। এবং যখনি কোনো নতুন আইনোড-ব্লক ব্যবহার করা হয়, বা কোনো ব্যবহৃত আইনোড ফাঁকা হয়, সেই আয়ব্যয়ের হিশেবটা সঙ্গে সঙ্গে তুলে দেওয়া হয় এই মেমরিতে তুলে নেওয়া সুপারব্লকের অ্যাকাউন্টস-এ। ঠিক ডেটা-ব্লকের বেলাতেও তাই। নতুন কোনো ডেটা-ব্লক যেই লেখা হয়, বা পুরোনো কোনো ডেটা-ব্লক ফাঁকা হয়, সঙ্গে সঙ্গে সেটাকেও কারনেল লিখে ফেলে মেমরিতে তোলা সুপারব্লকের কপিতে। মাঝে মাঝে, কিছু সময় অন্তর অন্তর, কারনেল তার মেমরিতে তুলে রাখা সুপারব্লকের প্রতিলিপির সঙ্গে পার্টিশনের সুপারব্লককে মিলিয়ে নেয়, মেমরিতে সুপারব্লকের সক্রিয় প্রতিলিপিটায় নতুন নতুন যা হিশেবে যোগ হয়েছে সেটাকে লিখে ফেলা হয় পার্টিশনের ফাইলসিস্টেমের সুপারব্লকে। এর নাম সিংক করা (sync) ... *খেজুরগাছে* ... । ঠিক সিংক করার আগের মুহূর্তটায় ভাবুন, মূল পার্টিশনের সুপারব্লকটা কিন্তু মেমরিতে সুপারব্লকের কপির চেয়ে পুরোনো। কিন্তু মেমরির সুপারব্লকটা সবসময়ই নতুন। সিস্টেমে শেষবার এই সিংক করাটা ঘটে যখন শাটডাউন ঘটে, মেমরির জ্যান্ত সুপারব্লকটাকে পার্টিশনের সুপারব্লকে লিখে কারনেল বেরিয়ে আসে। আপনার বাড়ির লোকও শান্তি পায়, যাক, আজকের মত মেশিন অফ হয়েছে। সুপারব্লকে যদি কোনো গোলযোগ থাকে, সিস্টেম তাহলে বুট করতে পারেনা, সেইজন্যে গ্নু-লিনাক্সে ডিস্কের বিভিন্ন জায়গায় একাধিক সুপারব্লক লেখা থাকে। যদি একটা সুপারব্লক ঘেঁটেও যায়, সিস্টেমকে আদেশ দেওয়া যায় অন্য সুপারব্লক ব্যবহার করার। গ্নু-লিনাক্সের পূর্বপুরুষ ইউনিক্সে এই বন্দোবস্ত ছিলনা।

১১.২।। আইনোড-ব্লক

সুপারব্লকের পরেই আসে আইনোড-ব্লক। একটা পার্টিশনের ফাইলসিস্টেমের প্রতিটি ফাইলের জন্যে একটা করে আইনোড রাখা থাকে এই আইনোড-ব্লক এলাকায়। এই আইনোড-ব্লক এলাকাটা কোনো ব্যবহারকারীর ঢোকার জন্যে নয়, শুধু কারনেলের জন্যে সংরক্ষিত। আইনোড-ব্লক এলাকায় ফাইলসিস্টেমের যাবতীয় ফাইলের আইনোডগুলো থাকে পরপর, পরস্পর-সন্নিহিত বা কন্টিগুয়াস রকমে। প্রত্যেকটা ফাইলের এই আইনোডে সেই ফাইল সম্পর্কে এই চরাচরে যা যা জানার থাকতে পারে তার মোটামুটি সবই লেখা থাকে। শুধু একটা জিনিষই থাকেনা এই আইনোডে, সেটা হল ফাইলের নাম। আর, ব্যবহারকারীদের প্রতি গ্নু-লিনাক্সের অপরিসীম যত্ন থেকে আরো দু-একটা জিনিষ ইচ্ছে করেই বাদ দেওয়া হয়, যেমন আপনি কোনো বেআইনি কাজের উদ্দেশ্যে ফাইলটা লিখছেন কিনা সেটা অনুল্লিখিত থাকে, গ্নু-লিনাক্স সিস্টেম খুব নিরাপদ — আগেই বলেছি না।

প্রতিটি ফাইলের এই আইনোড দৈর্ঘে একশো আঠাশ বাইটের একটা তালিকা। এর মধ্যে যা লেখা থাকে সেগুলো হল — এক, ফাইলের রকম, মানে, সেটা রেগুলার ফাইল না ডিরেক্টরি ফাইল না ডিভাইস ফাইল ইত্যাদি। দুই, কটা লিংক আছে ফাইলটার — মনে করতে পারছেন তো '`ls -al`' মেরে পাওয়া দীর্ঘ তালিকার থেকে — পরপর এখন মনে করে যান। তিন, ইউজারের ব্যক্তিগত পরিচিতির সংখ্যা বা ইউআইডি (UID)। চার, ইউজারের গ্রুপের পরিচিতিবাচক সংখ্যা বা জিইউআইডি (GUID)। পাঁচ, ফাইলের বিভিন্ন ধরনের অনুমতির ওই তিন তিরিখ্খে নটা রকম। ছয়, ফাইলে ধৃত মোট বাইটের সংখ্যা। সাত, ফাইলে ধৃত তথ্য শেষ কত তারিখ কোন সময়ে বদলানো হয়েছিল। আট, শেষ কত তারিখ কোন সময়ে ফাইলটা খোলা হয়েছিল। নয়, শেষ কত তারিখ কোন সময়ে আইনোডটা বদলানো হয়েছিল। দশ, ফাইলের পনেরোটা পয়েন্টারের একটা তালিকা। এই পয়েন্টারগুলো ফাইলের কতকগুলো চিহ্নক, ফাইলটায় যারা পৌঁছে দেয়, এদের নিয়ে আলোচনা আসার কোনো সম্ভাবনা নেই আমাদের এই জিএলটি ইশকুল পাঠমালায়, যদি না অনেকটা অব্দি আমরা সি এবং গ্নু-লিনাক্স কারনেলের সিস্টেম কল নিয়ে আলোচনায় আসি, যার সম্ভাবনা অত্যন্ত কম, এটা আট নম্বর দিন চলছে, শূন্য থেকে সাত। গোটাটা দশটা আলোচনায় শেষ করার প্ল্যান আছে। এর পরে আপনি নিজেই এগোতে পারবেন, যদি চান।

অক্টোবরের শেষ থেকে সেই বোকার মত লিখে যাচ্ছি। দু মাসের উপর হয়ে গেল। আজ তিরিশে ডিসেম্বর, এই প্রচন্ড ঠান্ডা, কাল ছবি দিয়েছিল, একটা গাবদু লালগাল বাচ্চা বরফের উপর দিয়ে হাঁটছে দার্জিলিঙে। সেটা দেখেও শীত করছিল। এই যে লিখছি, এই সকাল দশটাতেও আঙুলে যন্ত্রনা করছে, কিবোর্ড থেকে আঙুল সরে যাচ্ছে, টাইপের স্পিড কমে গেছে অনেক। মাঝে মাঝেই গরম চায়ের কাপে আঙুল ঘষে স্বাভাবিক করতে হচ্ছে। এই ভারখয়ানক্সে বসে লেখার প্রোজেক্টে গতকাল আমার কোটা ছিল সারাদিনে পৌনে চার লিটার চা, এবং গোটা পঁচিশেক সিগারেট। এদিকে পা-ভাঙার ছুটিটা নাকি উইদাউট পে হচ্ছেই, লেখার জন্যে আমার এত কামাই হয়। বইমেলা অপরের লেখা দিতে হবে শনিবার, আর এতো পারা যায়না, তার উপরে পয়েন্টার। ওসব নিজে পড়ুন। দরকার হলে জিএলটির সিডি নিয়ে যান, কত পড়বেন পড়ুন, পড়ে যান, আপনারা অনেকেই যে স্পিডে পড়েন, দু-এক দশকে শেষ হবার সম্ভাবনা নেই।

সিস্টেম আইনোডকে চেনে বোঝে ব্যবহার করে আইনোড নম্বর দিয়ে। এই আইনোড নম্বরটা একটা ফাইলসিস্টেমের প্রতিটি ফাইলের নিজস্ব এবং স্বতন্ত্র। এই নম্বরটা আসলে আইনোড-ব্লকের সমগ্রে ওই বিশেষ আইনোডটার অবস্থানের সূচক বা ইন্ডেক্স। আইনোড কথাটা যেখান থেকে এসেছে — ইন্ডেক্স নোড। তাই এই নম্বরটা থেকে নিছক যোগ বিয়োগ করে একটা আইনোডকে পেয়ে যায় সিস্টেম। আইনোড নম্বর কী ভাবে পেতে হয়, মনে আছে, '`ls`'-এর সঙ্গে '`-i`' অপশান হিশেবে দিয়ে। আপনি এখন সিস্টেমে নিজে ঘোরাফেরা করার জন্যে যথেষ্ট জানেন, সবকিছু নেড়ে চেড়ে দেখতে থাকুন, আর যেখানেই কোনো কিছু বুঝতে পারছেন না, সিস্টেম ডকুমেন্টেশন পড়ুন। তাতেও অনেকটাই বুঝতে পারবেন না, এই একটা ফাঁক, ওই একটা অস্পষ্টতা, কিছুটা আন্দাজ, এইরকম। তারপর একসময় দেখবেন, হঠাৎ করে গ্নু-লিনাক্স সিস্টেম তার আভ্যন্তরীন প্রক্রিয়া নিয়ে আপনার সামনে হাজির হতে শুরু করেছে। আমি এই গোটা পাঠমালাটায় যে চেষ্টাটা করছি। এই সিস্টেম বহু মানুষের বহু চেষ্টায় একটু একটু করে তৈরি হয়েছে, যা যা বানিয়েছে তারা তার গোটাটার পিছনে একটা জ্যান্ত চিন্তাপ্রক্রিয়া ছিল, সিস্টেমকে বোঝা মানে বানিয়েছে যারা তাদের চিন্তার ধরনটাকে বোঝা। আমি চাইছি আমি যেভাবে সেই প্রক্রিয়াটাকে বোঝার চেষ্টা করেছি তার কিছু প্রম্পট, কিছু কিউ, কিছু সূত্র আপনাদের সামনে ধরে নিতে — যাতে আপনারা পিছনের জ্যান্ত প্রক্রিয়াটাকে চিনতে শুরু করতে পারেন। আগেই তো বলেছি, এটা গ্নু-লিনাক্সের টেক্সট নয়, গ্নু-লিনাক্স ভাবা অভ্যাস করার প্র্যাকটিশ। যুক্তি-তক্কো-গপ্পোর পরও আজ কতগুলো বছর হয়ে গেল, এখনো ভাবা প্র্যাকটিস হলনা আমাদের। তখন দেশভাগের ব্যথা ছিল, এখন তো আর দেশভাগও হয়না, গোটা দেশটাই বিদেশ হয়ে যায় চোখের সামনে।

এখানে একটা জিনিষ খেয়াল করুন, আইনোডে কিন্তু ফাইলের নামও থাকেনা আবার আইনোডের নম্বরও থাকেনা। এই দুটোই রাখা থাকে ডিরেক্টরি ফাইলে। মনে আছে? যে ফাইল কেউই দেখতে পায়না, ডিরেক্টরির মধ্যে কী আছে সেটুকু অব্দিই দেখতে পায়। ফাইল পিছু ওই পনেরোটা পয়েন্টার ছাড়া আর একটা ফাইল সম্পর্কে আর সবকিছুই জানতে পারা যায় 'ls' কমান্ডের সঙ্গে বিভিন্ন অপশানে। নিজে খুঁজে দেখুন। এই 'ls' কমান্ডটা আমরা যখন চালাচ্ছি সে আসলে ওই ডিরেক্টরি ফাইল পড়েই আমাদের সামনে খুঁটিনাটিগুলো ফুটিয়ে তুলছে, সিস্টেম তো পড়তেই পারে ডিরেক্টরি ফাইল, আগেই বলেছি। যখনই আমরা একটা ফাইল খুলি, ঠিক ওই সুপার-ব্লকে যেমন, সিস্টেম হার্ডডিস্ক থেকে এর আইনোডটাকে তুলে নেয় মেমরিতে রাখা আইনোড টেবিলে। কারনেল গোটা সময়টাই কাজ করে মেমরিতে রাখা ওই আইনোডের কপি বা প্রতিলিপি নিয়ে। এই প্রতিলিপিটাকে ডাকা হয় 'আইনোড বাফার' বলে। বাফার নিয়ে আমাদের আলোচনা মনে করুন, এই আইনোড কপিটাও এই মুহূর্তে একটা বাফার। যাকে নির্দিষ্ট সময় অন্তর মিলিয়ে নেওয়া হয় ডিস্কে থাকা মূল আইনোডের সঙ্গে এবং নতুন গজিয়ে ওঠা পার্থক্যটুকু লিখে ফেলা হয় মূল আইনোডে। আবার সেই 'সিংক'। এবার ভাবুন, হঠাৎ করে যদি কারেন্ট অফ করে দেওয়া হয়, সিস্টেমের কাছে গোটাটা তো গোলমেলে হয়ে যাবে, তার মেমরির কপি আইনোড আর মূল আইনোডের মধ্যে একটা ফারাক রয়ে যেতে পারে। এই রকম সব কেরোসিন পরিস্থিতিতে ফাইলসিস্টেম চেক করার একটা ব্যবস্থা থাকে, সিস্টেম নিজেই সেটা করে নেয়, আলাদা করেও করা যায়। কমান্ডটা হল ফাইলসিস্টেম-চেক 'fsck'। প্রত্যেক আলাদা আলাদা ধরনের ফাইলসিস্টেমের জন্যে এর আলাদা আলাদা প্যাকেজ কাজ করে, 'fsck.ext2', 'fsck.reiserfs', 'fsck.vfat' ইত্যাদি। ...*খেজুরগাছে*..। শুধু একটা জিনিষ ভয়ানকভাবে মনে করিয়ে দিই — একটা পার্টিশনে মাউন্ট করা অবস্থায় কদাচ তাতে 'fsck' চালাতে নেই, কদাচ নয়, গোটা হার্ডডিস্কটাই ভোগে চলে যেতে পারে। এখনো আপনি মাউন্ট আর

মাউন্টের কনফিগারেশন ফাইল '/etc/fstab' জানেন না, তাই ম্যান বা ইনফো বা হাউটু পড়ে জানুন, কিন্তু সিস্টেম ঠিকমত বুঝে ওঠার আগে 'fsck' প্রয়োগ করতে যাবেন না, কিছুতেই না। আমাদের জিএলটির দেবাশিসের হার্ডডিস্কে আস্ত আস্ত পার্টিশন ব্যাড-সেক্টর বলে ঘোষিত হয়ে গেছিল।

১১.৩।। ডেটা-ব্লক

আমরা এর আগে যখন পার্টিশন সিলিন্ডার ট্র্যাক সেক্টর এলবিএ এইসবের আলোচনার সূত্রে 'লজিকাল-ব্লক' কথাটা উল্লেখ করেছি, তখন তার সংজ্ঞাটা আমরা জানতাম না, জানা সম্ভবও ছিলনা। এখন সেটা আমরা বুঝব। লজিকাল-ব্লক বা যৌক্তিক ব্লক বলে একটা ব্লককে আলাদা করা হয় ব্লকের ভৌত ধারণা বা ফিজিকাল-ব্লক থেকে। সচরাচর কোনো সিস্টেমে হার্ডডিস্ক কন্ট্রোলার ফাইল পড়া বা লেখার সময় তথ্য নাড়াচাড়া করে পাঁচশোবারো বাইটের এক একটা এককে। আইও ডিভাইস কন্ট্রোলার ব্যাপারটা, যদি ভুলে গিয়ে থাকেন, মনে পড়িয়ে নিন দুই নম্বর দিন থেকে। এই পাঁচশো বারো বাইটের প্যাকেটকে সচরাচর ডাকা হয় ফিজিকাল ব্লক বলে। কিন্তু কারনেল তার তথ্য লেখা বা পড়ার কাজে এই একক ব্যবহার করেনা। সেই ব্লককে ডাকা হয় লজিকাল বা যৌক্তিক ব্লক বলে। আমাদের সিস্টেমের কাজের মান অনেকটাই নির্ভর করে এই লজিকাল-ব্লকের আয়তনের উপর, কতটা দ্রুত বা শ্লথভাবে গোটা কাজটা সামলাবে সিস্টেম। এই ব্লক সাইজ বাড়ানো বা কমানোর অন্য একটা সমস্যার কথা আমরা উল্লেখ করে এসেছি একটু আগেই, পার্টিশনের আলোচনায়। মনে করতে পারছেন? এখুনি সেটা আবার আসবে।

আমরা আগেই বলেছি, দুই নম্বর দিনে, পরেও এসেছে, আমাদের ক্যারেকটার ডিভাইসগুলো, যেমন কনসোল বা প্রিন্টার, এরা তথ্য নাড়াচাড়া করে চিহ্ন বাই চিহ্ন, মানে ক্যারেকটারের এককে। কিন্তু সিস্টেম হার্ডডিস্কের মত ব্লক ডিভাইসগুলোর সঙ্গে তথ্য নাড়াচাড়া করে খাবলায়। এক খাবলা, আর এক খাবলা, এইরকম। এই খাবলাগুলোই হল ব্লক। এই লজিকাল ব্লক সাইজ সিস্টেম থেকে সিস্টেমে বদলায়। ১০২৪ বাইট থেকে ৮১৯২ বাইট বা হয়তো তারও বেশি। আমরা আগেই বলেছি, গ্নু-লিনাক্সে লজিকাল-ব্লক হল ১০২৪ বাইট। একটু আগে দেওয়া ১০.৩ নম্বর সেকশনে 'fdisk -l' কমান্ডের থেকে পাওয়া তালিকার সঙ্গে এর ঠিক পরেই দেওয়া ছবিটার পার্টিশন সাইজগুলো মেলান — প্রত্যেকটার ব্লক সংখ্যাকে ১০২৪ দিয়ে গুণ করুন, এতে আপনি মোট বাইট সাইজ পেলেন পার্টিশনটার, একে ১০২৪ দিয়ে ভাগ করলে কেবি, তাকে ১০২৪ দিয়ে ভাগ করলে এমবি, তাকে ১০২৪ দিয়ে ভাগ করলে জিবি। প্রত্যেকটা পার্টিশনকেই মেলান, মিলছে? এবার ভাবুন, একবার একটা লেখা বা পড়ার কাজে একটা ব্লক ব্যবহার করা মানে, আপনি হয়ত পাঁচ বাইট মাত্র তথ্য লিখলেন, তার জন্যে খরচ হয়ে গেল একটা গোটা ব্লক মানে ১০২৪ বাইট। এই দুর্মূল্যের বাজারে এক হাজার উনিশ পিস বাইট সিম্পলি নেই হয়ে গেল। কিন্তু আছে দুঃখ আছে মৃত্যু বিরহদহন লাগে, তবুও শান্তি তবু আনন্দ তবু অনন্ত জাগে। একটা সুসংগঠিত সিস্টেম, কোথাও কোনো ঘাপলা নেই, নিট কাজ করে যাচ্ছে — সে যে কী অপার শান্তি, মুহুর্মুহু হ্যাং নেই, একটা ফাইল বন্ধ করতে গিয়ে স্ক্যানডিস্ক চালিয়ে দেওয়া নেই, ইত্যাদি এবং ইত্যাদি।

এই ডেটা-ব্লকরা শুরু হয় আইনোড-ব্লকের পরে। ছবিটা দেখুন। প্রথমে বুট, তারপর সুপার, তারপর আইনোড, তারপর ডেটা। একদম এক নম্বর দিনের সেই আলোচনা মনে করুন, হার্ডডিস্কের প্রতিটা ফাইলের প্রতিটা তথ্যের একটা নিজের ঠিকানা আছে — সেই ঠিকানাটা হল মোট ডেটা-ব্লকের সমগ্রে একটা বিশেষ ডেটা-ব্লকের নিজস্ব অবস্থান সূচক নম্বর থেকে। মানে অমুক পার্টিশনের অমুক নম্বর ডেটা-ব্লক এই ঠিকানা দিয়ে আমরা ব্লকটাকে চিনি। যে ব্লকগুলোয় তথ্য আছে, তথ্য লেখা হয়ে গেছে, তাদের ডাকা হয় ডিরেক্ট ব্লক বলে। মনে হচ্ছেনা, হঠাৎ এর নাম 'ডিরেক্ট' কেন? এর মানে কী? কোন ব্লকগুলো ডিরেক্ট নয়? ঠিক এটার কথাই বলছিলাম। আপনি এখন সিস্টেম যারা বানিয়েছে তাদের চিন্তার গতিটা বোঝার চেষ্টা করছেন। আমি তপন বিশ্বাস বলে একটা ছেলেকে জানতাম, সে সাপ আর ভূতে ভরা অন্ধকার মসজিদ আর মাঠ দিয়ে হাঁটত আর দিকচক্রবাল দূরের ছোট্ট ছোট্ট আলোকিত জানলার চৌখুপি দেখে ভাবত, ওর মধ্যে লোকগুলো কী ভাবে? চেনা মানে তো চিন্তাকে চেনা।

ঠিকানা তো হল, পরপর সংখ্যা দিয়ে ব্লকদের সূচিত করা, কিন্তু, হারগিস একটা ফাইল পাওয়া যায়না, আরো সেই ফাইল যদি তুলে না-রেখে দিয়ে ব্যবহার করা হয়, যাতে পরপর সন্নিহিত বা কন্টিগুয়াস রকমে ব্লকগুচ্ছে তথ্য তোলা থাকে। ধরুন একটা ফাইল বাড়ছে, আপনি বাড়াচ্ছেন, ফাইলটার ঠিক শেষ ব্লকটার পরের ব্লকটা ফাঁকা নেই, তখন কারনেল কী করবে? তার কোনো দোষ নেই, সে গোটা ডিস্ক জুড়ে ছড়ানো ব্লকদের থেকে যেই কোনো ফাঁকা ব্লক

পায় সেখানেই লিখতে থাকে। কিন্তু এই হুডুম-দাডুম ছিটিয়ে থাকা ব্লকে বিশ্বময় ছড়িয়ে যাওয়া আপনার ফাইলকে পড়া বা লেখাটা ক্রমশই কারনেলের কাছে আরো আরো সময়সাধ্য হয়ে ওঠে, রিড-রাইট মানে লেখা-পড়ার গতি কমে যায়। এরই নাম ডিস্ক ফ্র্যাগমেন্টেশন। আবার দেখুন, এই ফ্র্যাগমেন্টেশন আছে বলেই ইচ্ছেমত ফাইল বাড়ানো বা কমানো যায়। এবার আপনি হয়ত প্রবল আবেগে আপনার কবিতার আরো একটা লাইন ভাবলেন, সিস্টেম আপনাকে জানালো, খবর — পরস্পরসন্নিহিতব্লকাভাবের অনিবার্য কারণবশত, আজ কাব্যলেখা বন্ধ আছে।
তার মানে, সবসময়ই এই হিশেবটা রেখে চলতে হবে, কোন কোন ব্লক ডিরেক্ট ব্লক। তাদের ঠিকানাগুলো তুলে রাখে আইনোড। কিন্তু তাদের সবার ঠিকানা আইনোড তুলে রাখতে পারেনা, মাত্র বারোটা অব্দি পারে — সেই স্থানাভাব। এবং এইজন্যে থাকে কিছু ইনডিরেক্ট ব্লক। এদের নিজেদের কোনো তথ্য থাকেনা। শুধু থাকে সেইসব ডিরেক্ট ব্লকের ঠিকানা যাদের আর আইনোডে রাখা যায়নি, মানে আইনোডের বারোটার পর তেরো নম্বর ব্লক থেকে।

দাঁড়ান একটা মজার কথা বলে নিই। ঠিক এর আগের লাইনে 'বারোটার' শব্দটা ছিল এই জিএলটি ইশকুল পাঠমালা সিরিজের একলাখতম শব্দ। এটা কি বিধির বিধান? এক লাখ শব্দে আপনাদের গ্নু-লিনাক্স শেখার বারোটা বাজিয়ে দিলাম?

আইনোডেই লেখা থাকে এই ইনডিরেক্ট ব্লকদের ঠিকানা। এবার আপনি বুঝতে পেরেছেন 'ডিরেক্ট' শব্দটার মানে কী?

এই সেকশনটা এখানেই শেষ করা যাক, ঘড়িতেও সকাল বারোটা আঠাশ বাজে, লাখশব্দ হয়েছে কেবল, পরের দিনই করা যাবে সেই গ্নু-লিনাক্স ফাইলসিস্টেম হায়েরার্কি। সেটা প্রায় সাদা-চামড়া মার্কিনীদের গুলিতে মৃত্যুমান শী-এনদের স্বপ্নের কানাডার মত না হয়ে যায়, শেষ ডিসেম্বরের বরফ জমা পাউডার নদী পেরিয়ে যেখানে তারা পৌঁছতে চেয়েছিল, কোনোদিনই পারেনি, হোক রেড ইন্ডিয়ান, এক একটা মানুষ আর কত গুলি নিতে পারে, এক একটা উপজাতি, এক একটা সভ্যতা? দুদিন বাদেই নববর্ষ, এখন তো কলকাতাতেও মার্কিন নববর্ষ হয়, পৃথিবী জুড়ে একটাই নববর্ষ, আমেরিকার নববর্ষ। কোনো এলিয়েন আক্রমন আসতে হয়নি, ইন্ডিপেন্ডেন্স ডে সিনেমার মত, এমনিতেই আমেরিকার ইন্ডিপেন্ডেন্স ডে গোটা পৃথিবীর ইন্ডিপেন্ডেন্স ডে করে দেওয়ার বিল পুলম্যানের সেই স্বপ্ন সত্যি হয়ে গেছে। কলকাতাতেও হয় মার্কিন নববর্ষ। মার্কিন নববর্ষ মানেই গোটা গোটা জাতির সভ্যতার ইতিহাসের অবশেষহীন অবলোপের সেলিব্রেশন। জাগো, কে কোথায় আছো, জাগো, তিন নম্বর বিশ্বের শুয়োরের কুকুরের এবং ছাগলের বাচ্চারা, মার্কিন নববর্ষের লেফটওভার বিতরণ হবে, জাগো, জেগে থাকো। পরের সেকশনের লেখাটাও ধরা যাবে নতুন বছরে আপনারা জেগে এবং চেগে যাওয়ার পরেই।

glt-mad@ilug-cal.org

MADHYAMGRAM

সংকলন ও রচনা : মধ্যমগ্রাম জিএলটি-র (glt-mad@ilug-cal.org) তরফে : ত্রিদিব সেনগুপ্ত

গ্নু-লিনাক্স ইশকুল

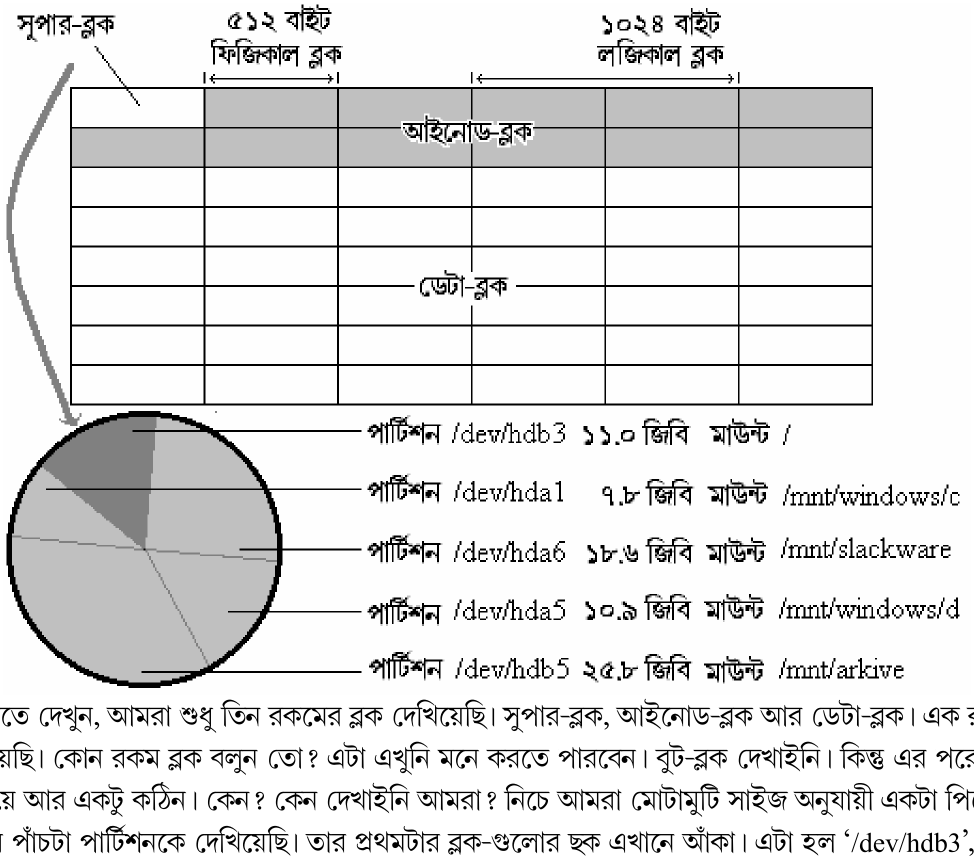
জিএলটি মধ্যমগ্রাম
glt-mad@ilug-cal.org

আজ পাঁচই জানুয়ারি, এখন বেলা বারোটাতেও হাড় কেঁপে যাচ্ছে, আরো আমার লেখার টেবিলের পাশেই দুদুটো জানলা দিয়ে উত্তর দিকের পুকুরের হাওয়া। তবে শীতই একমাত্র কারণ নয় ফের লেখায় ফেরার দেরির, এর মধ্যে এই সিরিজেরই তিন নম্বর দিন মানে চার নম্বর চ্যাপ্টারটাকে অপর বইমেলা সংখ্যার জন্যে আলাদা একটা প্রবন্ধ হিশেবে ছাপতে দিলাম, আর আলাদা করে নেওয়া মানেই কিছু টুকরোটাকরা মেরামত করতেই হয়। আর ওই করতে গিয়ে শ্যাননের 'এ ম্যাথামেটিকাল থিওরি অফ কমিউনিকেশনস' নেটে পেয়ে গেলাম, সেটা পড়বার লোভটা সামলাতে পারছিলাম না। যাকগে আবার আমি ফেরত এসেছি আমাদের গ্নু-লিনাক্স গোষ্ঠীর পিণ্ডদানে, যাকে ভদ্রলোকের ভাষায় বলে 'পেয়িং ব্যাক টু দি কমিউনিটি' — সেই ইয়ং বেঙ্গল আর বেম্মো মিশনারিদের সময় থেকে ভদ্রলোকরা তো ইংরিজিতে ছাড়া বোঝেনা।

।। দিন আট।।

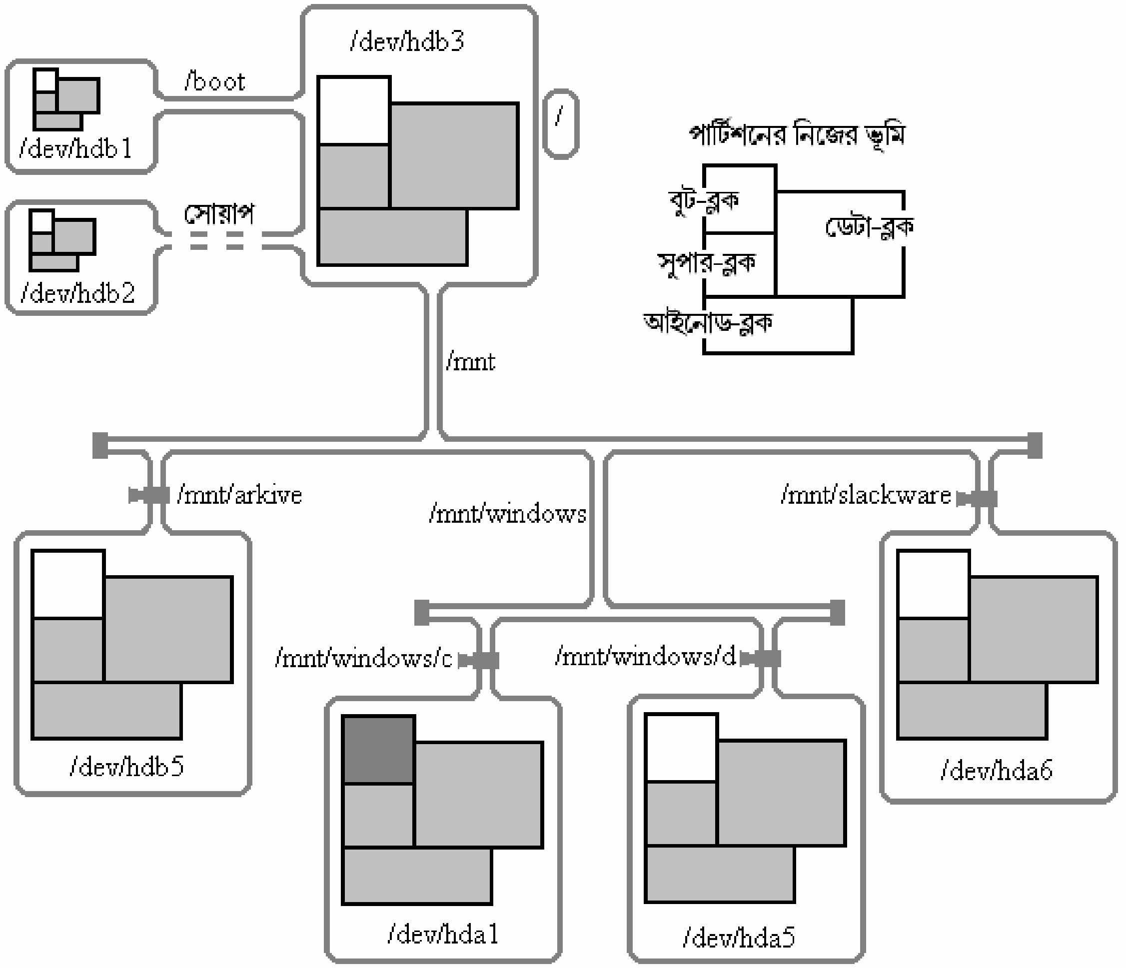
১।। ডিরেক্টরি আর কারনেল

সাত নম্বর দিনে যে আলোচনাটা হয়েছিল, সেটা খুব বড় হয়ে যাচ্ছিল বলে মাঝপথে দুম করে থামিয়ে দিয়েছিলাম, একটু মনে পড়িয়ে দিই। আমরা একটা গ্নু-লিনাক্স সামগ্রিক ব্যবস্থা বা ঐক্যবদ্ধ ফাইলসিস্টেমের এক একটা অংশ হিশেবে এক একটা পার্টিশনের এক একটা আলাদা ফাইলসিস্টেমের খুঁটিনাটি আলোচনা শুরু করেছিলাম, যে পার্টিশনগুলোর এক একটা আলাদা আলাদা ডিরেক্টরিতে মাউন্ট করি আমরা। — শেষ বাক্যটায় দুবার 'ফাইলসিস্টেম' কথাটা দুটো অর্থে ব্যবহৃত — মনে পড়ছে? সেটা করতে গিয়ে আমরা বিভিন্ন ধরনের ব্লকের কথা বলছিলাম, আর একটু মনে পড়িয়ে দিই, একটা ছবি দিয়ে।



এই ছবিতে দেখুন, আমরা শুধু তিন রকমের ব্লক দেখিয়েছি। সুপার-ব্লক, আইনোড-ব্লক আর ডেটা-ব্লক। এক রকম ব্লক বাদ দিয়েছি। কোন রকম ব্লক বলুন তো? এটা এখুনি মনে করতে পারবেন। বুট-ব্লক দেখাইনি। কিন্তু এর পরের প্রশ্নটা এর চেয়ে আর একটু কঠিন। কেন? কেন দেখাইনি আমরা? নিচে আমরা মোটামুটি সাইজ অনুযায়ী একটা পিঠের পাঁচ টুকরোয় পাঁচটা পার্টিশনকে দেখিয়েছি। তার প্রথমটার ব্লক-গুলোর ছক এখানে আঁকা। এটা হল '/dev/hdb3', যেখানে

মূল সুজে সিস্টেমটা থাকে, যার ডিরেক্টরি '/', এই পার্টিশনে কোনো বুট ব্লক থাকবে না, মানে বুট-ব্লক ফাঁকা থাকবে। এই পাঁচটা পার্টিশনের একটাতেই মাত্র বুট-ব্লক থাকবে। কেন, মনে করতে পারছেন? ভাবুন একটু। পরের মাউন্ট নিয়ে ছবিটা একবার দেখুন। এই ছবিটা এখন পুরোটা বুঝতে পারবেন না, এই ছবিটার কথায় আমরা আবার আসব। উপরে দেখুন আমরা ফিজিকাল ব্লক আর লজিকাল ব্লক দুটোকেই দেখিয়েছি। এখানে পাঁচটা পার্টিশনের পাঁচটা মাউন্টপয়েন্ট দেখিয়েছি, পাঁচটা ডিরেক্টরি। মাউন্ট ব্যাপারটার কেজো ছকে আপনারা ইতিমধ্যেই পরিচিত হয়েছেন, কিন্তু মাউন্ট-এর গোটা ব্যাপারটা এখনো আমাদের আলোচনায় আসেনি। একটু বাদেই আসব। আর একটা জিনিষ মনে রাখবেন, ছয় এবং সাত নম্বর দিনে দেওয়া আমার মেশিনের পার্টিশন টেবিলের তালিকার সঙ্গে মেলান, দেখবেন, আমরা দুটো পার্টিশনকে বাদ দিয়েছি, সোয়াপ পার্টিশন মানে '/dev/hdb2', আর সুজে সিস্টেমের বুট ফাইল যেখানে থাকে, সেই বুট ডিরেক্টরিতে যে পার্টিশন মাউন্ট করা হয়, মানে '/dev/hdb1'। এরা অন্যগুলোর তুলনায় বড্ড ছোট বলে। আর খেয়াল রাখুন, নিচে যেটা আমরা একটা পিঠের টুকরো করে দেখিয়েছি, আর একটু বাস্তব করতে চাইলে সেটাকে দুটো পিঠের টুকরোয় দেখানো উচিত, কারণ, হার্ডডিস্ক দুটো। পার্টিশনের নামগুলো থেকে খেয়াল করুন। আর, এখানে পিঠেটাকে আমার মেশিনের মোট ডিস্কভূমি বলে ভাবুন। যার মধ্যে বাস্তবে দুটো হার্ডডিস্ক আছে।



এবার আসুন আমরা একটা ডিরেক্টরিকে বোঝার চেষ্টা করি। ডিরেক্টরি ফাইলের অপৌরুষেয় বিদঘুটেপনা নিয়ে আমরা আগেই বলেছি, কারনেল ছাড়া কারুর পড়ার অধিকার নেই ডিরেক্টরি ফাইল, মায় রুটেরও না, আর সবাই শুধু দেখতে পারে সেই ডিরেক্টরিতে কী আছে, তাও যদি তার সেই ডিরেক্টরি দেখার অনুমতি থাকে, আর রুটের তো সব ডিরেক্টরি দেখারই অনুমতি থাকে। একটা ডিরেক্টরিতে একটা ফাইলের দুটো ব্যাপার লিপিবদ্ধ থাকে, এক, ফাইলটার নাম। আর দুই, তার আইনোড নম্বর। কেন আইনোড ব্লকগুলোর তালিকায় ফাইলের নামটা থাকেনা, সাত নম্বর দিনের

আলোচনায় আমরা বলেছি, এবারে বুঝতে পারছেন? ডিরেক্টরির কাছে যেহেতু একটা ফাইলের নাম আর আইনোড নম্বর দুই-ই লেখা থাকে, সে এবার সহজেই ওই আইনোড নম্বর দিয়ে ফাইলটাকে পেয়ে যেতে পারে। ডিরেক্টরি ফাইল আর আইনোড-ব্লক এই দুইয়ে মিলে একটা পরস্পর-সম্পর্কিত যৌথ-তালিকা তৈরি করে। আইনোড নম্বর হল এই দুটো তালিকার মধ্যে সম্পর্কটা। একটা ফাইলের সমস্ত কিছু যদি আমরা জানতে চাই — খুব সহজ, ফাইলটার আইনোড নম্বর দিয়ে দুটো তালিকারই ওই বিশেষ নম্বর লাইনটা পড়ে ফেলো।

সাত নম্বর দিনের আলোচনায় আমরা কী দেখলাম — যখনই একটা ফাইলের একটা লিংক বানানো হচ্ছে, তার আইনোডে উল্লিখিত লিংকের সংখ্যাটা এক করে বেড়ে যাচ্ছে। আর এই লিংক ফাইলটার নতুন ফাইলনামের জন্যে ডিরেক্টরিতেও একটা নতুন এন্ট্রি তৈরি হচ্ছে। যার আইনোড নম্বরটা নিট কপি করে দেওয়া হচ্ছে মূল ফাইলটার থেকে, যার সে লিংক। আপনার বানানো লিংকগুলোর যে কোনো একটাকে এবার '`rm`' করে দেখুন, লিংকটা উড়ে যাবে, আর তার সঙ্গে সঙ্গে '`ls -il`' করে বানানো আইনোড-সহ দীর্ঘ তালিকায় মূল ফাইলটার লিংক-সংখ্যা এক কমে গেল। এখন আমরা জানি এর মানে কী — আইনোড-ব্লকে লেখা লিংক-সংখ্যাকে এক কমিয়ে দেওয়া হয়েছে। আর ডিরেক্টরি ফাইলে এই লিংকটার জন্যে বানানো এন্ট্রিটাও উড়িয়ে দেওয়া হয়েছে। একটা ফাইলের শেষ লিংকটা উড়িয়ে দেওয়া হলে, তখন ফাইলটাই মুছে গেল। তার ডেটা-ব্লকগুলো ধুয়ে মুছে পোস্কার করে রাখা হল, নতুন ফাইল লেখার জন্যে। আপনার সঙ্গে এই পৃথিবীর শেষতম লিংক তো আপনি নিজেই, এটা কখনো খেয়াল করেছেন? আমাদের মতো যাদের ছদ্মনামে লেখার অভ্যেস তৈরি হয়, তাদের কাছে এটা খুব স্পষ্ট। ত্রিদিব সেনগুপ্তর লেখা সামনে এনে ধরে দিতে পারে কেবল একমাত্র দীপঙ্কর দাশই, ত্রিদিব সেনগুপ্ত হল দীপঙ্কর দাশের লিংক ফাইল। ত্রিদিব সেনগুপ্ত মুছে গেলেও দীপঙ্কর দাশ থাকবে। কিন্তু দীপঙ্কর দাশ নিজে দুনিয়ার ডিস্ক থেকে মুছে গেলে, আর সে নিজেই নিজের লেখাকে হাজির করতে পারবে না। তার জায়গায় অন্য ফাইল এলো, এখন থেকে অন্য কেউ অপরের লেখা দিতে দেরি করায় সোমনাথের কাছে খিস্তি খাবে।

এবার, কারনেলের কাছে, সিস্টেমের কাছে, একটা ফাইলের সমস্ত হালহকিকত, সমস্ত ঠিকানা কী ভাবে থাকে, সেটা আমাদের কাছে স্পষ্ট। এবার ধরুন আপনি আপনার তৈরি 'onefile' ফাইলটায় কী আছে দেখতে চাইলেন। আপনি তার কমান্ড জানেন, দিলেন কষে '`cat onefile`'। সামনে তো স্ক্রিন জুড়ে গোটা গোটা অক্ষরে 'onefile' ভেসে উঠল। কিন্তু স্ক্রিনকা পিছে কেয়া হ্যায়? কারনেল এই যে 'onefile' ফাইলটাকে খুঁজে পেতে সামনে এনে আপনার স্ক্রিনে মানে স্ট্যান্ডার্ড আউটপুটে ধরে দিল, তার মানে আগে তাকে 'onefile' ফাইলটার ঠিকানা খুঁজে পেতে হল, সেখান থেকে ফাইলটার শুরুর ব্লকের ঠিকানা আর শেষ ব্লকের ঠিকানা পেতে হল, শেষ অব্দি পড়ে নিতে হল, কারণ, এক স্ক্রিনে যদি শেষ অব্দি না-ধরে তাকে পরের স্ক্রিনে পরের স্ক্রিনে পরের স্ক্রিনে ... দেখিয়ে যেতেই হবে। এর জন্যে স্টেপ বাই স্টেপ তার কাজগুলোকে ভাবুন।

এক, আগেই আমরা বলেছি কারনেল কী ভাবে কারেন্ট ডিরেক্টরির, মানে এখন যেখানে আছেন, কাজ করছেন, সেই ডিরেক্টরির ডিরেক্টরি ফাইলের আইনোড নম্বরকে মেমরিতে তুলে নেয়। যদি মনে না পড়ে সাত নম্বর দিনের ১১ নম্বর সেকশনের সাবসেকশনগুলো একবার দেখে আসুন। এই আইনোড নম্বরটাকে কারনেল এবার খুঁজে বার করে আইনোড-ব্লকগুলো থেকে। সেখান থেকে এই ডিরেক্টরি ফাইলের আইনোডটা বার করে। দুই, এই ডিরেক্টরির আইনোড থেকে এবার কারনেল বার করে এই ডিরেক্টরি ফাইলের ডেটা-ব্লকের ঠিকানা। তিন, ডিরেক্টরি ফাইলের ডেটা-ব্লক পড়ে কারনেল এবার খুঁজে পায় 'onefile' ফাইলটাকে এবং পেয়ে যায় 'onefile' ফাইলটার আইনোড নম্বর। চার, 'onefile' ফাইলের আইনোড নম্বর পেয়ে যাওয়া মাত্রই কারনেল আবার ফেরত যায় আইনোড-ব্লকে। সেখানে গিয়ে 'onefile' ফাইলের আইনোড খুঁজে বার করে। পাঁচ, এই আইনোড থেকে কারনেল এবার পড়ে নেয় 'onefile' ফাইলটার আয়তন, এবং ডিস্কের শরীরে তার ডেটা-ব্লকের ঠিকানাগুলো। এখানে ইনডিরেক্ট ব্লক যদি থাকে তাহলে সেইটাও পড়ে বাড়তি ডেটা-ব্লকের ঠিকানাগুলোও পেয়ে যায়। ছয়, এবার কারনেল যায় তার ডিস্ক ড্রাইভারের কাছে। তাকে গিয়ে বলে তোমার ডিস্কের হেডগুলোকে নড়িয়ে তুমি তোমার প্ল্যাটারের গায়ে এত নম্বর সিলিন্ডারের এত এত সেক্টর থেকে এই এই ব্লকগুলোকে পড়ে ফেলো। এই হল ফাইলটার আয়তন, এই বাইটসাইজ হওয়া অব্দি তুমি পড়ে চলো। এইটা মিলে গেলে কারনেল বোঝে যে তার গোটা ফাইলটা পড়া হয়েছে, এবং তখন তার পড়ে ফেলা তথ্যটা ফুটিয়ে তোলার পালা, তার জন্যে সে যাবে ডিসপ্লে ড্রাইভারের কাছে। সে অন্য গল্প।

২।। সোয়াপ পার্টিশনের স্বতন্ত্র ফাইলব্যবস্থা

এতবার আলাদা আলাদা ভাবে দুটো অর্থে ব্যবহার করা হয়েছে যে আমরা এখন 'ফাইলসিস্টেম' শব্দটার এই দুটো ব্যবহারের সঙ্গে অভ্যস্ত। আমরা যখন '/' থেকে শুরু করে গোটা ডিরেক্টরিব্যবস্থাটার কথা ভাবছি, সেটা একটা ঐক্যবদ্ধ ফাইলব্যবস্থা বা ইউনিফায়েড ফাইলসিস্টেম। মাউন্ট করা বিভিন্ন পার্টিশনের ছবিটা, মানে আজকের দ্বিতীয় ছবিটা দেখুন, এর মধ্যে বিভিন্ন আলাদা আলাদা পার্টিশন মাউন্ট করা আছে বিভিন্ন আলাদা আলাদা ডিরেক্টরিতে। যেমন '/dev/hdb5' পার্টিশন আছে '/mnt/arkive' ডিরেক্টরিতে, '/dev/hda6' পার্টিশন আছে '/mnt/slackware' ডিরেক্টরিতে, আর '/dev/hda1' এবং '/dev/hda5' পার্টিশন দুটো মাউন্ট করা আছে '/mnt/windows' ডিরেক্টরির ভিতর '/mnt/windows/c' এবং '/mnt/windows/d' এই দুই সাবডিরেক্টরিতে। এই পার্টিশনগুলো এই এই ডিরেক্টরিতে মাউন্ট করা থাকতে পারে আবার নাও-পারে, এমনকি অন্য কোনো ডিরেক্টরিতেও মাউন্ট করা যেতে পারে, সেটা আমরা একটু বাদেই দেখব। কিন্তু পার্টিশন মাউন্ট করা থাকুক আর নাই-থাকুক, ডিরেক্টরিটা কিন্তু বানানো আছেই। সেটাই আমরা দেখিয়েছি যেন নলপথ আর তা আটকে দেওয়ার স্টপকক দিয়ে। যেন এক একটা পার্টিশনের এক এক টুকরো ডিস্কভূমি একটা আলাদা আলাদা রসদ। আলাদা আলাদা চৌবাচ্চায় আছে। তাদের সঙ্গে মূল চৌবাচ্চার যোগাযোগ ঘটে হায়েরার্কিকাল ডিরেক্টরি পথ বেয়ে, যা খোলা থাকে যখন পার্টিশনটা মাউন্টেড, আর যখন মাউন্টেড নেই তখন বন্ধ। মূল '/dev/hdb3' চৌবাচ্চাটা দেখুন '/' ডিরেক্টরি। তার সঙ্গে '/boot' ডিরেক্টরিতে '/dev/hdb1' পার্টিশন রয়েছে নলপথ দিয়ে সংযুক্ত। কিন্তু সেই নলে কোনো স্টপকক নেই, মানে এই পার্টিশনটা সবসময়েই মাউন্টেড আছে। আর এর ঠিক নিচেই রয়েছে '/dev/hdb2' পার্টিশন। কিন্তু তার কোনো নলপথ মানে ডিরেক্টরিপথ নেই। নলের দাগদুটো তাই, দেখুন, ভাঙা-ভাঙা আঁকা হয়েছে। এবং এর পাশে লেখা 'সোয়াপ'। এর মানে এই সোয়াপ পার্টিশনটাও সবসময়েই সংযুক্ত, কিন্তু সেটা একটা সম্পূর্ণ অন্যরকম ব্যবস্থা, স্বাভাবিক ফাইলব্যবস্থা বা ফাইলসিস্টেম থেকে একদম স্বতন্ত্র। এখানে আমরা ফাইলব্যবস্থা বলতে বোঝাচ্ছি ফাইল লেখা বা রাখা বা পড়ার রকম, যার নানা আলাদা আলাদা ব্যবস্থা হয়, আমরা আগেই বলেছি, ইএক্সটিটু রাইজারএফএস এক্সএফএস ইত্যাদি। কিন্তু সেই সবগুলো ফাইলব্যবস্থা এক দিকে, আর সোয়াপ ব্যবস্থা অন্য দিকে। সোয়াপ একদম আলাদা, তার কারণটা তো আপনারা জানেনেই, সোয়াপ ফাইলের জন্মই হয় মৃত্যুর জন্যে, নিজে মরে গিয়ে অন্য ফাইলকে বা ফাইলব্যবস্থাকে বদলে যাবে বলে, অস্থায়ী ভাবে স্মৃতিকে লিখে রাখার একটা জায়গা সোয়াপ-ফাইল, আমরা বারবার বলেছি — সোয়াপ ফাইল হল র‍্যামের সহচর, র‍্যামকে বাড়িয়ে তোলে। তাই এই পার্টিশনের ফাইল লেখা রাখা পড়ার কায়দা অন্যরকম হবেই।

তার মানে, একটা মূল পার্টিশন যা মাউন্ট হবে '/' ডিরেক্টরিতে, আর একটা সোয়াপ পার্টিশন — এই দুই রকমের দুটো পার্টিশনে দুই ধরনের ফাইল-ব্যবস্থা, এটা একটা গ্নু-লিনাক্স সিস্টেমের ন্যূনতম প্রয়োজনীয় শর্ত। এরপর কী কী পার্টিশন থাকবে কী কী ফাইলব্যবস্থায় সেটা আপনার ইচ্ছের এবং প্রয়োজনের উপর নির্ভর করছে। প্রথমটাকে সচরাচর বলা হয় রুট পার্টিশন আর দ্বিতীয়টাকে সোয়াপ পার্টিশন। এমনকি খুব বিদঘুটে বিতিকিচ্ছিরি একটা অবস্থায়, যাকে ইংরিজিতে বলে পরিস্থিতি, এমনকি এই সোয়াপ পার্টিশনও না-রাখা যায়। কিন্তু রুট পার্টিশন না-থাকা মানে সিস্টেমটাই না-থাকা। গ্নু-লিনাক্স একটা সিস্টেমে এই রুট পার্টিশনেই থাকে সিস্টেমের গোটা কাঠামোটা। রুট ডিরেক্টরি এবং তার থেকে বেরোনো সমস্ত সাবডিরেক্টরির কাঠামোটা মনে করুন, ছ নম্বর দিনে এসেছিল, যার কোথায় কী থাকবে সেই ছকটা, যার নাম লিনাক্স ফাইলসিস্টেম হায়েরার্কি, যেটা আপনাদের বারবার বলে আসছি, ওই সামনে, ওই সামনে, কিন্তু কিছুতেই আর এসে পৌঁছচ্ছি না। আসলে লেখার আগে আমি বুঝতেও পারিনি, এক একটা ছোট ছোট জায়গা বলে আসতে এতটা করে লিখতে হবে। যাকগে, এই রুট পার্টিশনের মধ্যে ওই হায়েরার্কিকাল ডিরেক্টরি ব্যবস্থাটা থাকে, থাকে সমস্ত অবশ্য প্রয়োজনীয় ফাইল এবং প্রোগ্রাম।

সোয়াপ পার্টিশন, যা অবশ্যই থাকা উচিত একটা সিস্টেমে, কাজ করার সুবিধের জন্যেই, মূলত কারনেলের কাজের সহায়তা করে। এক আর দুই নম্বর দিনে বিভিন্ন ধরনের মাল্টিপ্লেক্সিং-এর আলোচনার সূত্রে আমরা যে একত্র চলমান অগণ্য প্রসেস বা প্রক্রিয়ার কথা বলেছি, সেই প্রক্রিয়াগুলোর প্রতিটির গতিশীলতাকে, কাজকে, কাজ করতে গিয়ে সিস্টেমের নানা রসদের ব্যবহারকে চূড়ান্ত ভাবে নিয়ন্ত্রণ করে কারনেল। এই নিয়ন্ত্রণ এবং সীমিত রসদকে সুবিধেজনক ভাবে বিভিন্ন প্রক্রিয়ার মধ্যে ভাগবাঁটোয়ারার কাজে কারনেলকে সহায়তা করে সোয়াপ। অনেকগুলো সমান্তরাল

প্রক্রিয়ার ভারে সিস্টেম যখন ভারাক্রান্ত, সিস্টেমের মেমরির পিঠ যখন বেঁকে গেছে বোঝা তুলতে তুলতে তখন তার একটু দম নেওয়ার ব্যবস্থা করার জন্যে কারনেল কিছুটা বোঝা অস্থায়ী ভাবে নামিয়ে রাখে এই সোয়াপে। সোয়াপ শব্দটার আভিধানিক অর্থ একটা কিছুর সঙ্গে অন্য একটা কিছুকে বদলানো। এখানেও ঠিক তাই ঘটে, র‌্যামের একদম জ্যান্ত চলমানতা থেকে সোয়াপের অস্থায়ী আরামে পৌঁছয় একটা প্রক্রিয়া। তারপর যেই ফুরসত আসে, আরামে বদলানো প্রক্রিয়া আবার দৌড় করাতে শুরু করে কারনেল, সোয়াপ থেকে তাকে তুলে র‌্যামে ফেরত এনে। এই সোয়াপ পার্টিশনের এই বিশেষ ধরনের ফাইলব্যবস্থার বৈশিষ্ট্য হল এই যে একে সরাসরি কোনো ব্যবহারকারী পড়তে পারেনা, এমনকি রুট ইউজারও না। সোয়াপকে ব্যবহার করে শুধু কারনেল।

এবার, সিস্টেমে বসে আপনি সোয়াপ সম্পর্কে আর একটু কিছু জানতে চাইলে, তার উপায় কী? মনে আছে? প্রথমে দিন ‘`man -k swap`’। স্ক্রিনে অনেকটা লেখা ফুটে উঠলে তাকে ফাইল বানিয়ে পড়ার কোনো উপায় আছে? এরকম কিছু শুনেছেন কোথাও? যাকগে, এখানে সে সমস্যা নেই, সোয়াপটা মোটামুটি ভদ্রগোছেরই আছে। মাত্র আট দশটা লাইন। এখানে তার কয়েকটা লাইন তুলি।

```
mkswap(8)  - set up a Linux swap area
swapon(2)  - start/stop swapping to file/device
swapoff(2) - start/stop swapping to file/device
swapon(8)  - enable/disable devices and files for paging and swapping
swapoff(8) - enable/disable devices and files for paging and swapping
```

ওই গোটা দশেক লাইনের থেকেও কয়েকটা ছেঁটে দিলাম, এত বেশি টেকনিকাল সেগুলো। এবার দেখুন তো, নিজে পড়ে, কী আন্দাজ পাচ্ছেন কমান্ডগুলো সম্পর্কে? ‘mkswap’ স্বাভাবিক ভাবেই, সোয়াপফাইল বানায়, মানে যে পার্টিশনটাকে আপনি সোয়াপ পার্টিশন বানাতে চান, সেই পার্টিশনে সোয়াপ ফাইল লেখার বানানোর ব্যবহার করার মত করে নিজস্ব স্বতন্ত্র ফাইলব্যবস্থা বানিয়ে তোলে ‘mkswap’। সচরাচর এটা চালু প্রথা যে নতুন ব্যবহারকারীর গ্নু-লিনাক্স ইনস্টলেশন করে দেয় তার চেনাজানা কোনো লিনাক্সী। এমনকি আপনি নিজেও যদি ইনস্টল করে থাকেন, তাহলেও, ইনস্টল করার সময়, আপনি আসলে এই কমান্ডটাই ব্যবহার করেছেন, না-জেনে। ইনস্টলেশনের সময় যখন দেখিয়েছে, কোথায় সোয়াপ পার্টিশন রাখব, কতটা তার সাইজ রাখব, তখন আপনি চালু প্রথা মেনে, মোটামুটি র‌্যামের ডাবল, সাইজ দিয়েছেন, কোথায় বানাবে বলে দিয়েছেন, বা এমনকি আপনার হয়ে সিস্টেমই সেটা করেছে, নিজেই মোটামুটি একটা গড় ব্যবস্থা ছকে নিয়েছে, তারপর সেই অনুযায়ী ইনস্টল করেছে। সিস্টেম নিজেই করুক, অন্য কেউ করে দিক, আপনি করে থাকুন, শেষ অব্দি ব্যবহার হয়েছে এই ‘mkswap’। সামনে আপনি কোনো ছবিওলা ঝিনচাক দেখানোর দাঁত দেখেছেন — হলুদ লাল নীল ধূসর, কোনোটা ইএক্সটিটু, কোনোটা রাইজার, কোনোটা ফ্যাটথার্টিটু, কোনোটা পদিপিসির বর্মিবাক্স, এই রোগা হচ্ছে, এই মোটা হচ্ছে, এই রং পালটাচ্ছে। সেই ফ্রন্ট-এন্ডের পিছনে খাওয়ার দাঁত মানে ‘mkswap’ দিয়েই হার্ডডিস্ক ভূমিটাকে চিবিয়ে চিবিয়ে সোয়াপ ফাইল লেখার মত করে সাইজ করেছে সিস্টেম। আর কোন পার্টিশনে আপনি সোয়াপ বানাবেন সেটা সিস্টেমকে দিয়েছেন ‘swapon’ ব্যবহার করে, আপনি হয়ত ইঁদুরের পেট টিপেছেন, কিন্তু ব্যাক-এন্ডে ছিল ‘swapon’। তেমনি আপনি যদি কোনো পার্টিশনে সোয়াপ ফাইল আর না-রাখতে চান, সেটাকে অন্যকাজে ব্যবহার করতে চান, তখন, আপনার ইঁদুরের নাড়িভুড়ি বেয়ে দৌড় করেছে ‘swapoff’।

এবার আপনার এই ‘হয় হয় কিন্তু Zানতে না-পারা’ ক্যালিটা সরাসরি কমান্ড প্রম্পটে শো করে দিয়ে নিজের ইগোকে একটু পিঠ-চুলকে দিতে পারেন, কিন্তু, দেখবেন, ভালো করে পার্টিশন ছক না-বুঝে এইসব করতে গিয়ে সিস্টেমকে কাতুকুতু দিয়ে বসবেন না, গোটাটা নেড়ে ঘেঁটে গিয়ে একটা সমূহ কেলেংকারি। আর কমান্ডগুলো সম্পর্কে ভালো করে জানবেন কী করে? মরুঅঞ্চলের উদ্ভিদের বাংলার পলিমৃত্তিকায় কৃষির সাফল্য বিষয়ে গবেষণা নিয়ে কী একটা গান আমাদের কানে এসেছিল না?

ও, একটা কথার উত্তর দিন তো, এই তালিকাটায় ‘swapon’ আর ‘swapoff’ কমান্ডদুটোর দুবার করে দুটো আলাদা নম্বরের ম্যানপেজের কথা বলার মানে কী? দেখুন, দুটো কমান্ডই দুবার করে আছে একবার ‘2’ আর একবার ‘8’। তার মানে একটা দেখতে হবে ‘`man 2 swapon`’ বা ‘`man 2 swapoff`’ কমান্ড দিয়ে, আর একবার দেখতে হবে ‘`man 8 swapon`’ বা ‘`man 8 swapoff`’ কমান্ড দিয়ে। কিন্তু কেন? যদি না বুঝতে পারেন ছয় নম্বর দিনের

গোড়ার দিকটা একবার উল্টে আসুন। গোটা পুরোনো আলোচনাটা আপনার মাথায় একদম হাতেগরম থাকা চাই। সচরাচর টেক্সটগুলোয় যা হয়, ঠিক গঠনের নিয়ম মেনে প্রসঙ্গগুলো পরপর আসে, তাতে একটু বোরিং হয় ঠিকই, কিন্তু একটা সুবিধাও থাকে, এগোনোর পিছনোর, আটকে গেলে পিছিয়ে গিয়ে বুঝে-আসার। একদম বিষয়-নির্ভর বা অবজেকটিভ পাঠ। আর এই লেখাটা হচ্ছে তার বিপরীত ধরনে, বিষয়ী-নির্ভর বা সাবজেকটিভ পাঠ বলা যায়। যেন একজন জ্যান্ত কেউ সিস্টেমটা বুঝতে বুঝতে চলেছে, যেখানে যেখানে তার যেটুকু আটকে যাচ্ছে সেটুকু সে আলোচনা করে নিচ্ছে, ওই জ্যান্ত ব্যবহারকারীটা বেশি উপস্থিত ব্যবহারের বিদ্যাটার চেয়েও। তাই এখানে ব্যবহারকারীর দায়িত্ব একটু বেশি এসে পড়েই, গোটাটা নিজের মাথায় তুলতে তুলতে যাওয়ার। এখানে আটকে গেলে আপনি তো খুঁজেই পাবেন না প্রসঙ্গটা কোথায় এসেছে। ঠিক আমাকে যতটা গোটা লেখাটা মাথায় রাখতে হচ্ছে আপনাকেও তাই করতে হবে, আমি খেয়ে-না-খেয়ে লিখে চলেছি, আর আপনি চুয়িংগাম চিবোতে চিবোতে পড়ে ফেলবেন, সেটা চলবে না।

৩।। নানা ধরনের ফাইলব্যবস্থা

আগের সেকশনে আমরা বললাম রুট আর সোয়াপ — ফাইলব্যবস্থার এই দুটো বড় বিভাগের কথা। এবার এই রুট পার্টিশনের ফাইলব্যবস্থাটা ঠিক কী রকম হবে? তার নানা প্রকারভেদ আছে। সোয়াপ ফাইলব্যবস্থায় যা নেই। আমার মেশিনের একই সোয়াপ পার্টিশন সুজে আর স্ল্যাকওয়ার দুজনেই সমানভাবে ব্যবহার করে, যখন যেটা দিয়ে বুট করি। আগে যখন সুজে স্ল্যাকওয়ার আর রেডহ্যাট তিনটেই ছিল, তিনটেরই তাই। সোয়াপ ফাইলব্যবস্থার প্রকারভেদ হয়না। কিন্তু রুট পার্টিশনের হয়। আর এছাড়াও নানা পার্টিশন থাকে, নানা আলাদা আলাদা ফাইলব্যবস্থায়, আমরা আগেই দেখেছি। সিস্টেমের নিরাপত্তা, ব্যবহারের সুবিধার জন্যেও নানা পার্টিশন রাখতে হয়, আমরা বলেছি। আপনার গ্নু-লিনাক্স সিস্টেমে একটা বিশেষ মুহূর্তে মোট কত রকমের ফাইলব্যবস্থা ব্যবহৃত হচ্ছে, সেটা জানার একটা উপায় হল, ‘`cat /proc/filesystems`’।

এই ‘/proc’ ডিরেক্টরিটা আমরা ঐক্যবদ্ধ ফাইলসিস্টেমের ডিরেক্টরি কাঠামোর ছবিতে দেখেছি, এখানে কী থাকে সেটায় আমরা একটু বাদে আসব গ্নু-লিনাক্স ফাইলসিস্টেম হায়েরার্কির আলোচনায়। এই যে একটা বিশেষ মুহূর্তের কথা বললাম, সেটা খুব গুরুত্বপূর্ণ। ধরুন, আপনার সিস্টেমে একটা পার্টিশন আছে যেখানে ফাইল লেখার রাখার পড়ার ব্যবস্থাটা হল রাইজারএফএস, এবার সেই পার্টিশনটা আপনি সবসময় মাউন্ট করেন না, যখন দরকার পড়ে মাউন্ট করে নেন। আমার সিস্টেমে ‘/dev/hdb5’ আর ‘/dev/hda6’ পার্টিশনদুটো যেমন, আমার যখন দরকার পড়ে আমি মাউন্ট করে নিই ‘/mnt/arkive’ আর ‘/mnt/slackware’ ডিরেক্টরিতে। মাউন্ট কখন করবে, আপনাআপনি সিস্টেম একটা পার্টিশনে মাউন্ট করে নেবে কিনা — এসব নিজের মর্জিমত ঠিক করে নেওয়া যায়, ‘/etc/fstab’ বলে একটা ফাইল মারফত, মাউন্টের আলোচনায় আমরা সেকথায় আসব। এছাড়া আর কোনো পার্টিশনে রাইজারএফএস নেই। এবার, যখন আমার সিস্টেমে এই পার্টিশনদুটো মাউন্ট করা নেই, তখন আমি ‘`cat /proc/filesystems`’ কমান্ড দিলে ও কোনো রাইজারএফএস ওর তালিকায় দেখায় না, কিন্তু মাউন্ট করা থাকলে দেখায়।

এবার আমার সিস্টেমে সবগুলো পার্টিশন মাউন্ট করা অবস্থায় আমি ‘`cat /proc/filesystems`’ দিয়ে যে তালিকাটা পেলাম সেটা এখানে তুলে দিই। অবশ্যই এই রকম চার স্তম্ভে আমি পাইনি, সেটাকে করে নিতে হয়েছে ‘’ বলে একটা কমান্ড দিয়ে। ‘`cat /proc/filesystems|pr -4>filesystems`’ কমান্ড দিয়ে প্রথমে তালিকাটাকে পাইপ করা হয়েছে ‘pr’-এর কাছে, ‘pr’ তালিকাটাকে চার স্তম্ভে স্তম্ভিত অবস্থায় রিডাইরেক্ট করেছে ‘filesystems’ নামের ফাইলে। ‘pr’ আসলে প্রিন্ট সংক্রান্ত কমান্ড, আগে ‘lpr’ বলে একটা কমান্ড হত, লাইন প্রিন্ট করার কাজে ব্যবহারের, কিন্তু এই সব কাজে বেড়ে ব্যবহার করা যায়, পাতার নম্বর, কলাম, রো, হেডার, ফুটার, সব ঠিক করে দেয় একটা টেক্সট ফাইলে। নিজে ব্যবহার করে দেখুন। সেই ফাইল থেকে তুলে দিচ্ছি, একটু ছাঁটকাট করে। আপনারা যখন ম্যানপেজ পড়ছেন, তখন তাকেও রিডাইরেক্ট করে একটা ফাইলে এনে সেটাকে এইভাবে ‘pr’ দিয়ে ফরমাট করে নিতে পারেন, পড়ার সুবিধে হয়। এছাড়া ম্যানপেজ থেকে ওয়েবপেজ বা ‘*.html’ ফরম্যাট বা অন্য কোনো ফরম্যাটে বদলে নেওয়ার অনেক উপায় আছে। আমি যেটা ব্যবহার করি সেটা ‘rman’। ধরুন আপনি ‘ls’-এর ম্যানপেজ পড়তে চাইছেন। তাকে রিডাইরেক্ট করে ফাইল বানিয়ে নিতে পারেন ‘`man ls>lsman`’। একে ‘pr’

দিয়েও ফরমাট করে নিতে পারেন। বা ‘rman’ দিয়ে করলে, ‘`man ls|rman -f html>lsman.html`’। এতে ‘ls’ কমান্ডের ম্যানপেজটাকে ‘rman’ একটা ওয়েবপেজ করে দেবে যার নাম ‘lsman.html’। একে আপনি আপনার ইন্টারনেট ব্রাউজার দিয়ে পড়তে পারবেন। কমান্ড প্রম্পটে ব্যবহার করার জবর দুটো টেক্সট মোড ব্রাউজার হল ‘links’ আর ‘lynx’। আরো কী কী আছে, মনে করতে পারছি না। এই ব্রাউজারদুটো দিয়ে চাইলে ছবিসহ একদম স্বাভাবিক চেহারাতেও ব্রাউজ করা যায়, তখন আলাদা করে ‘`-graphics`’ অপশান দিতে হয়।

আমি তো অনেকসময়ই নেটে ব্রাউজ করি টেক্সট মোডে। এতে সুবিধে হল, প্রচুর গাম্বাট লোক তাদের ওয়েবপেজে গাড়লের মত দুম্বো দুম্বো সব গ্রাফিক্স আর জাভাস্ক্রিপ্ট দিয়ে রাখে, তার আর কী, খরচ তো আপনার, লোড হতে প্রচুর সময় লাগে। ‘links’ বা ‘lynx’ টেক্সট মোডে নিট দরকারি জিনিষটুকু এনে দেয়, দরকার পড়লে জাস্ট একটা ‘d’ মেরে ডাউনলোড করে নাও। আর আপনার ফরম্যাট করা ম্যানপেজগুলো বা যে ম্যানপেজের সংগ্রহ এমনিতেই দেওয়া আছে আপনার সিস্টেমে তাদেরও পড়তে পারেন দিয়ে। আপনি, কমান্ড প্রম্পটে থাকলে তো ভালোই, যদি এই মুহূর্তে এক্স-উইনডোজেও থাকেন, অসুবিধে নেই, একটা টার্মিনাল বা একটা কনসোল খুলুন, নেটে কানেক্ট করে কমান্ড দিন ‘`links www.geocities.com/ddipankardas/`’। আর যদি ব্যাকগ্রাউন্ডে জিএলটির লোগোর অনবদ্য ছবিটা সহ — আমি করেছি না? — পাতাটা দেখতে চান, তখন জাস্ট একটা ‘`-g`’ যোগ করে দিন কমান্ডটার সঙ্গে। ছবিদুটো কুচো সাইজের বললেও কম বলা হয়, ‘glt-logo.png’ আর ‘glt-bkg.png’ দুটোই দুই কেবির কম। এটা ইয়াহু জিওসিটিতে মধ্যমগ্রাম জিএলটির সাইট, বানানো এখনো শেষ হয়নি, উপরের কয়েকটা মাত্র লিংক কাজ করছে। এই লেখাটা শেষ হলে, তথাগত অশেষ সঙ্কর্ষণ ওদের সব পোকাবাছা হয়ে গেলে সবগুলো চ্যাপ্টারই থাকবে এখানে, পিডিএফে। ‘links’ কমান্ড দেওয়ার পরে যদি দেখায় ‘`command not found`’, তার মানে, আপনার সিস্টেমে ‘links’ ইনস্টল করা নেই, জাস্ট ইনস্টলেশন সিডিটা দিয়ে ইনস্টল করে নিন। ‘rman’-ও তাই। যাকগে, এবার আসা যাক ফাইলসিস্টেমের তালিকায়। ‘pr’ কি সুন্দর নিজে নিজেই আজকের তারিখ সময় আর পাতার নম্বর করে দিয়েছে দেখুন।

```
2004-01-07 11:07                                       Page 1

rootfs         tmpfs          minix          usbdevfs
bdev           shm            iso9660        usbfs
proc           pipefs         nfs            reiserfs
sockfs         ext2           devpts         vfat
futexfs        ramfs          xfs
```

এই ফাইলসিস্টেমের নামের তালিকা থেকে দেখুন, কয়েকটা নাম আপনার চেনা, অন্য কয়েকটা নয়। দু-চারটে কথা বলে নেওয়া যাক এদের নিয়ে। শুরু করা যাক গ্নু-লিনাক্সে সবচেয়ে বেশি ব্যবহৃত ডিফল্ট ফাইলসিস্টেমগুলো থেকে। কয়েক বছর আগেও এই বিষয়টা ছিল খুব সরল, ব্যবহারযোগ্য ফাইলব্যবস্থা গ্নু-লিনাক্সে দুই কি ম্যাক্সিমাম তিনের বেশি ছিলনা, মূলত ব্যবহার হত ইএক্সটিটু নিদেন রাইজারএফএস। এখন তা নয়, অনেকগুলো ফাইলসিস্টেম রেগুলার ব্যবহার হয়ে চলেছে অজস্র গ্নু-লিনাক্স সিস্টেমে। ইএক্সটিটু ক্রমে দিগন্তের ওপারে চলে যেতে শুরু করেছে। এমনকী পরিবর্তনের গড়িমসির জন্যে পরিচিত রেডহ্যাট লিনাক্সও তার ভার্শন নাইন থেকে ডিফল্ট করে দিয়েছে ইএক্সটিথ্রিকে। ম্যানড্রেক সুজে এগুলোকে তো ছেড়েই দিন। এমনকী বিশুদ্ধতাবাদী স্ল্যাকওয়ারও তার ইনস্টল সিডির কারনেলগুলোর ভিতর একটা এক্সএফএস কারনেলও দিয়ে দিচ্ছে। কারনেল ভারশন ২.৪ থেকে গ্নু-লিনাক্স অনেকগুলো ফাইলসিস্টেমকে ব্যবহার করার সুযোগ দেয়। তার অনেকগুলো ফাইলসিস্টেমের খুঁটিনাটিই দেখবেন দেওয়া আছে আপনার সিস্টেমে, ‘/usr/src/linux/Documentation’ ডিরেক্টরির মধ্যে। টেক্সট ফাইলে।

এখানে আমরা একটা ফাইলসিস্টেমেরও গঠনের খুঁটিনাটি জটিলতাতেও যাবনা, সে এক বীভৎস বিরাট ব্যাপার, এখানে কিছুটা দেওয়ার প্ল্যানে আমি অনেকটা পরিশ্রম করলাম কয়েকদিন ধরে। সঙ্কর্ষণও একটা ভালো লিংক দিয়েছে, আইবিএম সাইটে, লিনাক্স ফাইলসিস্টেম সংক্রান্ত। ড্যানিয়েল রবিনস-এর দশটা নিবন্ধের একটা সিরিজ, http://www-106.ibm.com/developerworks/library/l-fs.html, লিনাক্স ফাইলসিস্টেম নিয়ে। আর একটা খুব ভালো লিংক জুয়ান স্যান্টোস ফ্লোরিডোর লেখা, জারনাল ফাইল সিস্টেম নিয়ে, লিনাক্স গেজেটে। তার লিংকটা হল, http://www.linuxgazette.com/issue55/index.html। নেমসিস-এর (Namesys) নিজের সাইটে, মানে যাদের

রাইজারএফএস, রাইজারের উপর খুব ভালো একটা ডকুমেন্ট আছে। ঠিক এসজিআই-এ (SGI) যেমন আছে তাদের নিজেদের তৈরি এক্সএফএস নিয়ে। আমার ইচ্ছে ছিল বেশ কিছুটা খুঁটিনাটি নিয়ে এগুলোকে ধরা, শেষে নিজেরই কাপড়চোপড় খুলে যাওয়ার যোগাড়, একটাকে বুঝতে আর একটা, তাকে বুঝতে আর একটা, তার উপর খুব বেশি সময় নেটে থাকতে গেলে নিজেরই অপরাধবোধ হয় একটা, সংসার তো চালাতে হবে। আর এই লেখাটাও প্রায় শেষের দিকে চলে আসছে, খুব বেশি লম্বা করার ইচ্ছেও নেই। এখানে আমরা তাই খুব শর্টে সারব, জাস্ট এক একটা ফাইলসিস্টেমের সুবিধে অসুবিধে নিয়ে, কী কাজের উপযোগী, ইত্যাদি। একটা কথা মাথায় রাখবেন, নানা ধরনের ফাইলসিস্টেম লাগে নানা ধরনের কাজের জন্যে। এমন একটা ফাইলসিস্টেমও নেই যা সব কাজ সবচেয়ে ভালো ভাবে করতে পারে। সবচেয়ে নিরাপদ এবং সুস্থির ফাইলসিস্টেমেও প্রচুর গন্ডগোল তৈরি হয়, তাদের সামলাতে হয়, তাই সবসময়ই ব্যাকআপ করা উচিত নিজের তথ্যের, নিয়মিত ভাবে।

৩.১।। ইএক্সটিটু

গ্নু-লিনাক্স জন্মানোর একদম প্রথমতম ইতিহাস থেকেই প্রায় চালু ইএক্সটিটু (Ext2)। ইএক্সটিটু এসেছে তার আগেকার এক্সটেন্ডেড ফাইলসিস্টেম ব্যবস্থা থেকে, নামের আত্মীয়তাটা খেয়াল করুন। এক্সটেন্ডেড ফাইল সিস্টেম (Extended-File-System) সিনে এসেছিল এপ্রিল ১৯৯২-এ। কারনেল ভার্শন ০.৯৬সি-র সময়। এরপর বারবার বদলাতে থেকেছে। এই বদলের সূত্র ধরেই এসেছিল ইএক্সটিটু এবং গ্নু-লিনাক্সের সবচেয়ে জনপ্রিয় ফাইলসিস্টেম হয়ে উঠেছিল একটানা বেশ কয়েকবছর ধরে। জার্নালিং ফাইলসিস্টেম মানে জার্নাল ব্যবহার করে এমন ফাইলসিস্টেম আসার পরই ধীরে ধীরে সিন থেকে সরে যেতে শুরু করল ইএক্সটিটু। জার্নাল ব্যাপারটা নিয়ে বিশদ আলোচনার লিংক তো আগেই দিয়েছি, এই লেখাটা তার জন্যে খুব একটা উপযুক্ত জায়গাও নয়, এবং আমিও আলোচক হিশেবে আদৌ খুব উপযুক্ত নই, এত বেশি টেকনিকাল সেটা। তবু ব্যাপারটা কী সেটা একটু বলে নিই।

একটা ফাইলসিস্টেমের জার্নাল বলতে বোঝায় একটা ছক, একটা কাঠামো। এই ছকটাও রাখা থাকে ডিস্কের ওই পার্টিশনেই, যে পার্টিশনের ফাইলসিস্টেমকে জার্নাল করা হচ্ছে। ছক বা কাঠামোটা হল একধরনের একটা নথি বা লগ, যেখানে সমস্ত বদলগুলো লিপিবদ্ধ হয়। ইনফ্যাক্ট বলা ভাল, বদলগুলো হওয়ার আগেই লিপিবদ্ধ হয়। ফাইলসিস্টেম তার নিজের মেটাডেটায় কোনো বদল ঘটানোর আগে সেই বদলের হিশেবনিকেশ এই জার্নালে লিখে রাখে। মেটাডেটা বলতে বোঝায় একটা ফাইলসিস্টেমের আভ্যন্তরীন তথ্যের কাঠামো। যে কাঠামোটার কাজ এটা নিশ্চিত করা যে তথ্যের এবং তথ্যের সজ্জাটার থোবড়া বিলা হয়ে যায়নি, হুলিয়া বদলে যায়নি, ঘেঁটে যায়নি সেগুলো। এই 'মেটা' কথাটা উত্তরআধুনিক সংস্কৃতিচর্চারও একটা খুব প্রিয় টার্ম। মেটান্যারেটিভ, মেটাটেক্সট — ন্যারেটিভ বিষয়ে ন্যারেটিভ, টেক্সট বিষয়ে টেক্সট। আমি অনেকদিন আগে 'নিরন্তর প্রব্রজ্যায়: দ্বিতীয় খসড়া' নামে একটা উপন্যাস লিখেছিলাম, সেটা এই অর্থে মেটানভেল। একটা উপন্যাস কী করে লিখব তাই নিয়ে একটা উপন্যাস। যে কোনো কিছুতেই এটা ঘটতে পারে। কবিতা নাটক ফিল্ম। এখানে মেটাডেটা সেই রকম ডেটা বিষয়ে ডেটা, তথ্য বিষয়ে তথ্য। প্রত্যেক ফাইলসিস্টেমেরই নিজের একটা মেটাডেটা থাকে, এবং মেটাডেটা লেখার একটা রকম থাকে। মেটাডেটার কাঠামোর তফাত একটা ফাইলসিস্টেম থেকে আর একটা ফাইলসিস্টেমের তফাতের একটা বড় জায়গা। মেটাডেটাকে বদলে যেতে না-দেওয়া — একটা ফাইলসিস্টেমের জরুরি কাজগুলোর মধ্যেও সবচেয়ে জরুরি। কারণ, ভেবে দেখুন, মেটাডেটা বদলানো মানেই আপনার সিস্টেম আর ওই পার্টিশনের ওই ফাইলসিস্টেমে রাখা তথ্যের সজ্জাকে কাঠামোকে তথ্যকে চিনতে পারবেনা। তথ্যগুলো তখন আছে এবং নেই। মানে নেই। ওরকম সন্ধ্যাভাষার অস্পষ্টতা নিয়ে সাইকোলজি চলে, কাব্য চলে, হার্ডডিস্কের প্ল্যাটার আর হেড চলেনা।

জার্নালাইজড ফাইলসিস্টেমের জার্নালের কাজ এই মেটাডেটায় কোনো বদল ঘটানোর আগে লিখে রাখা। সিস্টেম বুট করার সময় ফাইলসিস্টেম যে চেক হয়, বা কোনো গন্ডগোল ঘটলে, বিদ্যুৎবিভ্রাট ইত্যাদি, যখন চেক করা হয়, আগেই বলেছি আমরা, সেই চেকের বা নিরীক্ষার কাজে সময়ের প্রয়োজনটা ভয়ানকভাবে কমিয়ে দেয় জার্নাল। এমনিতে ভাবুন, প্রতিটি সিলিন্ডারের ট্র্যাকের কাঠামো ধরে ধরে তথ্য এবং তথ্যের সজ্জাকে মেলানোর কাজটা কমে গিয়ে দাঁড়াচ্ছে জাস্ট জার্নাল মেলানোর কাজে। গোটা ডিরেক্টরি সিস্টেম ফাইল সিস্টেমটা আর মেলাতে হচ্ছেনা। শুধু জার্নালটা আর একবার চালিয়ে দেওয়া হচ্ছে। এই জার্নাল ব্যবহার করা ফাইলসিস্টেম আসার পরেই ইএক্সটিটু ফাইলসিস্টেমের জনপ্রিয়তা কমে যেতে শুরু করল। আসলে জার্নাল ব্যবহার করাটা ফাইলসিস্টেম চেকের সময়টাকে

এমন অবিশ্বাস্য ভাবে কমিয়ে দেয়। কিন্তু এখনো একটা বিরাট সংখ্যক লিনাক্সীর কাছেই ইএক্সটিটু খুব প্রিয় একটা ফাইলসিস্টেম। তার একটা বড় কারণ এই যে, এটার সঙ্গে দীর্ঘদিনের অভ্যস্ততা ছাড়াও, এত বেশি মেশিনের এত বেশি সিস্টেমে এত বেশি ব্যবহারকারী ইএক্সটিটুকে ব্যবহার করেছেন যে এর মত বারবার পরীক্ষিত ফাইলসিস্টেম আর হওয়া সম্ভব নয়। স্বাভাবিক অবস্থায়, সিস্টেম অফ হওয়ার সময়, কারনেল সমস্ত বাফারকে তার নিজের নিজের জায়গায় সঠিকভাবে লিখে দিয়ে পার্টিশন থেকে বেরিয়ে আসে, মানে পার্টিশনটাকে নিজের সিস্টেমের বাইরে বার করে দেয়। একে বলে আনমাউন্ট করা। এক এক করে কারনেল প্রতিটি পার্টিশন থেকে আনমাউন্ট করে, এবং এই পক্রিয়ার শেষে সে নিজেই নিজের কাছ থেকে ছুটি নেয়, আমরা স্ক্রিনে ফুটে উঠতে দেখি

`Sending all processes the TERM signal...`

`Sending all processes the KILL signal...`, মেশিন অফ হয়।

বিদ্যুৎবিভ্রাট বা অন্য কোনো কারণে, ভুল ভাবে মেশিন বন্ধ করার কারণে, বা অন্য কোনো গোলযোগে, যখন স্বাভাবিক আনমাউন্ট ঘটেনা, মানে, সঠিকভাবে বাফারগুলোকে লিখে সমস্ত জরুরি তথ্যকে তার নিজের জায়গায় রেখে সঙ্গতভাবে বেরিয়ে আসতে পারেনা কারনেল, তখন পরের বার বুট হওয়ার সময় ফাইলসিস্টেম চেকিং শুরু হয়। আমরা 'fsck' কমান্ডটার কথা বলেছিলাম, এর ইএক্সটিটু ভার্শনটার নাম 'fsck.ext2'। সেই 'fsck.ext2' তখন চালু হয়। তার কাজ গোটা মেটাডেটাকে পুনঃসংস্থাপিত করা তার সঠিকতায়। এইটা করতে গিয়ে তাকে গোটা ফাইলসিস্টেমটাই তন্নতন্ন করে খুঁটিয়ে খুঁটিয়ে পরখ করতে হয়, শুধু মেটাডেটার বিটগুলোর সাম্প্রতিকতম বদলটাকে বিশ্লেষণ করলেই কাজ হয়না। 'fsck.ext2' বকেয়া থেকে যাওয়া তথ্য বা ডেটা-ব্লকগুলোকে একটা 'lost+found' নামের ডিরেক্টরিতে তুলে রাখে। তার মানে, প্রচুর প্রচুর সময়, জার্নালের লগটা চালিয়ে দেখতে যা সময় লাগে তার চেয়ে অনেক বেশি। বড় হার্ডডিস্কে এই কাজে দুতিন ঘন্টাও লেগে যেতে পারে। কিন্তু আবার জার্নাল রাখেনা বলেই মেটাডেটার পরিমাণটা অনেক ছোট হয়, মেমরি অনেক কম লাগে এই ফাইলসিস্টেমের ব্যবহারে। তাই ব্যবহারের সময়, যদি সিস্টেমের র‍্যাম খুব কম হয়, ইএক্সটিটু অনেক তাড়াতাড়ি কাজ করতে পারে অন্য অনেক ফাইলসিস্টেমের চেয়ে।

৩.২।। ইএক্সটিথ্রি

আপগ্রেড করা বা উন্নততর করে তোলাটা কম্পিউটার জগতে, হার্ডওয়ার আর সফটওয়ার দুটোতেই, খুব বড় একটা ইশু। ইএক্সটিটুকে ইএক্সটিথ্রিতে আপগ্রেড করে তোলার কাজটা খুবই সহজ। ইএক্সটিথ্রি (Ext3) আবার জার্নালিত ফাইলসিস্টেম। ইএক্সটিটুর সহজতা এবং জার্নালের দ্রুততা একসাথে পাওয়া যায়। ইএক্সটিথ্রির সঙ্গে অন্যান্য পরবর্তী যুগের জার্নালিত ফাইলসিস্টেমের তফাত এটাই যে ইএক্সটিথ্রি কিন্তু এক্সটেন্ডেড ফাইল সিস্টেমের কাঠামো থেকে সম্পূর্ণ বেরিয়ে যায়নি। নামের ধারাবাহিকতাটা দেখুন। ইএক্সটিথ্রি প্রায় সরাসরি ইএক্সটিটু-র উত্তরাধিকার বহন করে। দুটোর আত্মীয়তার জায়গাটা খুবই পোক্ত। আর এই কারণেই একটা ইএক্সটিটু ফাইলসিস্টেমের উপর খুবই সহজে একটা ইএক্সটিথ্রি ফাইলসিস্টেম বানিয়ে তোলা যায়। এদের একমাত্র গুরুত্বপূর্ণ তফাত ওই জার্নাল।

ইএক্সটিথ্রি-র সুবিধেগুলোর একটা, তাই, খুব সহজে ইএক্সটিটু থেকে বদলে নিতে পারা। ফাইলসিস্টেমের আভ্যন্তরীণ তথ্য এবং তথ্যসজ্জার কোনো গলতি হওয়ার সম্ভাবনা থাকেনা। ইএক্সটিটু-র কোডের উপরেই তৈরি ইএক্সটিথ্রির কোড, আর ডিস্কের ভিতর তথ্য রাখার কৌশল বা মেটাডেটার কাঠামো দুটো ফাইলসিস্টেমেই প্রায় হুবহু এক, তাই ইএক্সটিটুতে কোনো পার্টিশন ফরম্যাট করা থাকলে, এবং পরে সেটা বদলে কোনো জার্নালিত ফাইলব্যবস্থায় যেতে চাইলে, রাইজারএফএস (ReiserFS) এক্সএফএস (XFS) বা জেএফএস (JFS) জাতীয় অন্য কোনো সিস্টেমে যাওয়ার যে ঝামেলা সেটা ইএক্সটিথ্রির বেলায় হয়না। ওগুলোয় যাওয়ার আগে পার্টিশনের গোটা তথ্যের ব্যাকআপ করে নিতে হয়, এবং সেই ব্যাকআপ থেকে ফের শুরু করে গোটাটা বানিয়ে নিতে হয়। সেটা ইএক্সটিথ্রি-তে হয়না। আর ফের গোটাটা বানিয়ে নিতে গিয়ে কিছু ঘাপলা না-ঘটাটাই আশ্চর্য। কোনো একটা লিংক কাজ করছে না, কোনো একটা লাইব্রেরি পাচ্ছেনা, ইত্যাদি। এটা ইএক্সটিথ্রি-র বেলায় হয়না, কারণ আসলে ওই কাঠামোটা তো বদলাচ্ছেই না। আবার ইএক্সটিথ্রি থেকে কখনো ইএক্সটিটু-তে ফেরত যেতে চাইলে সেটাও দিব্য সহজ। একটা পরিচ্ছন্ন রক্তপাতহীন আনমাউন্ট করো ইএক্সটিথ্রি পার্টিশন থেকে, আবার তাকে রিমাউন্ট করো ইএক্সটিটু হিশেবে। আমরা দেখব একটু বাদেই, কী করে এই অপশানগুলো দিতে হয়। অন্য জার্নালিত ফাইলব্যবস্থাগুলোয় তথ্যসততা কায়েম রাখার কায়দাটা

হল শুধু মেটাডেটাকে নিয়ন্ত্রণ করে। মেটাডেটা সবসময়েই অনাঘ্রাত অনাহত থাকবে। কিন্তু মূল ফাইলব্যবস্থার অপরিবর্তনীয়তাটা এতটা বর্মচর্মাবৃত নয়। ইএক্সটিথ্রি-র বেলায় তা নয়, সে তথ্য এবং তথ্য-বিষয়ক-তথ্য, ডেটা এবং মেটাডেটা, দুয়েরই নিরাপত্তার দেখভাল করে। কতটা যত্ন নিয়ে দেখভাল করবে সেটাও বলে দেওয়া যায়, মানে নিজের মত করে বদলে কাস্টমাইজ করে নেওয়া যায়। ডেটা এবং জার্নালকে তুল্যমূল্য করে দিলে, 'data=journal' মোডে সবচেয়ে বেশি নিরাপত্তা, কিন্তু কাজ করে খুব আস্তে। এতে তথ্যের গায়ে কোনো আঁচড় পড়েনা, কারণ তথ্য এবং তথ্যের-তথ্য, ডেটা এবং মেটাডেটা, দুয়েরই জার্নাল রাখা হচ্ছে। বরং 'data=ordered' মোডে ডেটা এবং মেটাডেটা দুয়ের অটুটতাতেও নজর রাখা হয়, কিন্তু জার্নাল রাখা হয় শুধু মেটাডেটার। মেটাডেটার এক একবার বদলের সঙ্গে সম্পৃক্ত যাবতীয় ব্লকগুলোকে সংগ্রহ করে রাখে ফাইলসিস্টেম ড্রাইভার, এক একটা দলে, এক একটা 'ট্রানজাকশন'-এর ('transaction') আকারে। এবং মেটাডেটায় বদল ঘটানোর আগে এক একটা গোটা ট্রানজাকশনকে লিখে ফেলা হয় হার্ডডিস্কের শরীরে। এতে কাজের গতিও ঠিক থাকে, আবার তথ্যের নিরাপত্তাও বজায় থাকে। আর একটা মোড হল 'data=writeback'। এই মোডে মেটাডেটার খাতায় বদলের নথী লিখে ফেলার পর হার্ডডিস্কের মূল ফাইলসিস্টেমে তথ্য লেখা হয়। মোডের নামটা দেখুন, আগে-পরে-টা বুঝতে পারবেন। এই তিন-নম্বর মোডে কাজ সবচেয়ে তাড়াতাড়ি হয়। কিন্তু একটা কোনো বিদ্যুৎবিভ্রাট বা ওই জাতীয় কোনো কেলোর পরে একটা বিদঘুটে ব্যাপার ঘটার সম্ভাবনা থাকে। আপনার কবিতার ফাইলে যেসব লাইন আপনি ইতিমধ্যেই উড়িয়ে দিয়েছেন, সেগুলো ফের পেয়ে যেতে পারেন ফাইলে। কেন ভাবুন তো? আভ্যন্তরীন ফাইলসিস্টেমের তো কোনো হানি হয়নি, সেটা অটুট আছে। এবার, আপনি যে বদল ঘটিয়েছিলেন, সেটা মেটাডেটায় লিখে ফেলার, রাইটব্যাক করার আগেই কেলোটা ঘটেছিল। তাই আপনার বর্জিত লাইনেরা আবার কোরাসে গাইতে শুরু করেছে, লাইনেই ছিলাম বাবা। যদি নিজের থেকে কোনো পছন্দ না দিয়ে দেন, তাহলে ইএক্সটিথ্রি চলে তার ডিফল্ট সেটিং-এ, মানে 'data=ordered', মানে শুরুতে খানিক মৃণাল-মাণিক অন্তে সুভাষ ঘাই দিয়ে কোনাকুনি বনসাইয়ে সাজানো টোনা-টুনির নিরাপদ সংসার।

৩.৩।। রাইজারএফএস

কারনেল ভার্শন ২.৪ থেকে আনুষ্ঠানিক ভাবে, অফিশিয়ালি পাওয়া গেল, কিন্তু তার কিছু আগে থেকেই রাইজারএফএস একটা কারনেল প্যাচের আকারে পাওয়া যেত। প্যাচ বলতে একদম তাপ্পি বলতে আমরা যা বুঝি। শুধু নেগেটিভ নয়, পজিটিভ তাপ্পি। মূল প্যান্টুলুনের ফুটো ঢাকতে তাপ্পি মারার কথা না-ভেবে ভাবুন একটা বাড়তি পকেট পাওয়ার আশায় পজিটিভ তাপ্পি। প্যাচকে মূল কাঠামোর গায়ে সেঁটে দেওয়া হয় বাড়তি কিছু পাওয়ার আশায়। বাড়তি কোনো কাজ, বাড়তি কোনো সুবিধা। ল্যারি ওয়াল আর পল এগার্টের এই প্যাচের সফটওয়ার ধারণাটা বেশ ইন্টারেস্টিং। কিন্তু এখানে সেটা আলোচনার জায়গা নয়। কারনেলের মধ্যে যা নেই, কিন্তু বাইরে থেকে যোগ করে নেওয়া যেতে পারে, কারনেলের ভার্শন স্থির হয়ে যাওয়ার আর যা কাজ হয়েছে সেগুলো দিয়ে দেওয়া যায় এই প্যাচে। একটু '`man patch`' করে নিতে পারেন। প্যাচ বুঝতে গেলে আবার একটু ডিফ (diff) বুঝতে হবে, ডিফ পড়ে দেখুন, দুটো ফাইলের তফাত বা ডিফারেন্স নিয়ে কাজ করে, খুব মজার।

রাইজার যাদের তৈরি সেই কোম্পানির নাম নেমসিস-কে (Namesys) খুব সহজেই নেমেসিস পড়া যায়, কিন্তু গ্রীক পুরাণ মাফিক তথ্যের কোনো সমুহ বিনাশ আদৌ এর উদ্দেশ্য নয়, বরং তার উল্টোটা, রাইজারএফএস এসেছিল পুরোনো ইএক্সটিটু-র একটা শক্তিশালী বিকল্প হিশেবে, তথ্যের শক্তিশালীতর যত্নআত্তির আশায়। সীমিত ডিস্কভূমির আরো কাবেল ব্যবহার, আরো কম জায়গায় মোট আরো তথ্য রাখা, রাইজারের একটা জোরের জায়গা। এবং ক্র্যাশ বা কেলো থেকে দ্রুততর আরোগ্য, যার কথা আমরা আগেই বলেছি। কিন্তু একটা সমস্যার জায়গা এই যে রাইজারের বেশি মনোযোগ ডেটার চেয়েও মেটাডেটার দিকে। পরে হয়ত রাইজার শুধু মেটাডেটা নয়, ডেটারও জার্নাল রাখতে শুরু করবে, ইএক্সটিথ্রির বেলায় যেমন বলছিলাম।

ডিস্কভূমির খুব ভালো ব্যবহার হয় রাইজারে। এই কাজে রাইজারএফএস মূলত ব্যবহার করে ব্যালান্সড-ট্রি অ্যালগরিদম। শুধু রাইজারএফএস নয়, এই ব্যালান্সড ট্রি অ্যালগরিদম ব্যবহার করে এইচএফএস (HFS), এনএসএস (NSS) এবং স্পাইরালগ (Spiralog) ফাইলসিস্টেমও। যদিও রাইজারএফএস ভার্শন ফোর, ব্যালান্সড ট্রি-র জায়গায় ড্যান্সিং ট্রি অ্যালগরিদমের কথা বলেছে। কিন্তু এটা এমন একটা জায়গায় ঢুকে পড়ছি, আমি ঢোকার চেষ্টা করে দেখেছি, অন্তত এখনো আমার সেখানে কথা বলার যোগ্যতা নেই। আমি আপনাদের এই রাইজার, এরপর জেএফএস

আর এক্সএফএস-এর একটা হালকা উপরসা আন্দাজ দিয়ে বেরিয়ে যাব। যতটুকু কথা না-জানলে অন্য খুব সাধারণ কাজগুলোও বুঝতে অসুবিধে হয়। কাজ করতে গিয়ে আমি দেখেছি, রাইজারএফএস আর এক্সএফএস অনেক সুবিধাজনক, রাইজার ব্যবহারের পরামর্শ আমায় দিয়েছিল অরিজিৎ, আর এক্সএফএসেরটা দিয়েছিল তথাগত। এবং ইএক্সটিটু বা ইএক্সটিথ্রির সঙ্গে গতির পার্থক্য এতটাই যে একবার ম্যানড্রেক ৮.২-এ এক্সএফএসে অভ্যস্ত হয়ে যাওয়ার পরে রেডহ্যাটে ইএক্সটিটু বা থ্রিতে, বিশেষ করে কিছু কম্পাইল করার সময়, আমার কান্না পেয়ে যেত। রেডহ্যাট উড়িয়েই দিয়েছিলাম। শেষে ইনস্টল করার সময়ে দ্বিতীয় ভার্চুয়াল কনসোলে গিয়ে 'mkfs.reiserfs' ব্যবহার করে পার্টিশনটা রাইজার বানিয়ে এক নম্বর কনসোলে ফেরত এসে পার্টিশন আর ফরম্যাট না-করে ইনস্টল করে, রাইজারে রেডহ্যাট করেছিলাম, তবে সিস্টেম ভদ্রস্থ হয়েছিল। রেডহ্যাটে এক্সএফএস করার কোনো উপায় আমি পাইনি, তবে ইন্দ্র বলেছিল ওর কাছে কী একটা সিডি আছে যা দিয়ে এক্সএফএসে করা যায়, আমায় দেবে, তা ইন্দ্র যা ব্যস্ত থাকে, ফোন করবে এবং ফোন করল এই দুটোর মধ্যে থাকে একমাসের দূরত্ব, যে আমি সামনের জন্মের আগে ওটা পাওয়ার খুব বেশি আশা করিনা। আমার নিজের অন্তত এক্সএফএস সবচেয়ে পছন্দের, কিন্তু স্ল্যাকওয়্যার নাইনে আমায় এক্সএফএস থেকে রাইজারে ফেরত যেতে হয়েছে, কারণ এক্সএফএস কারনেল যেটা থাকে সেটার সঙ্গে কিছু কিছু কারনেল মডিউলের বেশ সমস্যা হয়েছে। হতে পারে আমার এনভিডিয়া জিফোর্স কার্ডের জন্যে। কারণ, জিফোর্সের ড্রাইভারে দেখেছি কিছু সমস্যা হয়। আমার স্ল্যাকওয়্যার সিস্টেমে এক্সএফএস কারনেলে, এমপ্লেয়ার আর জিএসএল মানে গ্নু সায়েন্টিফিক লাইব্রেরি, সি-তে গণিত আর বিজ্ঞানের কাজ করার একটা লাইব্রেরি, কিছুতেই কম্পাইলড হচ্ছিল না। রাইজারে ফেরত আসার পর আবার সব চমৎকার হয়ে গেছে। দেখি, স্ল্যাকওয়্যার নাইন পয়েন্ট ওয়ানের সিডি ইমেজ ডাউনলোড করছে তথাগত, কালই ফোন করে জানিয়েছে, অনেকটাই হয়ে গেছে, যদি সেটায় ভালো ভাবে এক্সএফএসে করা যায় তাহলে আমার সবচেয়ে পছন্দের ডিস্ট্রো স্ল্যাকওয়ারে সবচেয়ে পছন্দের ফাইলসিস্টেম এক্সএফএসে কাজ করতে পারব। একই সঙ্গে এটাও বলে রাখা ভাল এই পছন্দগুলো কোনো রকমেই সব জেনে বুঝে পছন্দ নয়। হতেই পারে অন্য কোনো কিছু আমার আরো পছন্দ হত একটু বিশেষ রকমে বদলে নিলেই, যাকে টেকনিকাল ভাষায় বলে 'টুইক করা'। সেটার কথা আদৌ জানিই-না, এত কম জানি কম্পিউটারের। তবে, চারদিকের অবস্থা যা, কালকেই যাদবপুরে অংকুর বাংলা সিডি নিয়ে গ্নু-লিনাক্স সেমিনারের গল্প শুনছিলাম। সেখানে একজন সিনিয়র সিটিজেন কোনো একটা মন্তব্যের সূত্রে বলেছেন, আমি তো 'লিনাক্স এইট' ব্যবহার করি। এবং এই ঘটনাটার উপাখ্যানতার মূল খোঁচটা এইখানে যে লোকটা যাদবপুর কম্পিউটার সায়েন্স ডিপার্টমেন্টের ফ্যাকাল্টি মেম্বার, সেখানে পড়ায়, এবং সে, নাকি, লিনাক্স নিয়ে কাজ করে। এটা যদি আমি বলতাম, আমাদের জিএলটি-র নতুন ছেলেদের কেউ বলত, সেটা অবাক লাগত, এমা কিছুই তো জানিস না — লিনাক্স এইট বলেই কিছু হয়না। কিন্তু এই পরিস্থিতিতে এটা একটা অপদার্থ ধৃষ্টতা, একটা স্পর্ধা, একটা জাতিসত্তার ধর্ষণ। ইংরেজরা আমাদের যেমন চাইত তেমনি হয়ে থাকা। চারপাশটাই এইরকমই — একটা শক্তির কেন্দ্র, তার চারদিকে ছোট ছোট বোকা বোকা জমিদাররা। সমস্ত বিশ্বাস মূল্যবোধ সবকিছুই, ঠিক ওই লোকটার লিনাক্সে কাজের মতই, একটা জনশ্রুতি কিম্বদন্তী হিয়ারসে — 'নাকি' কাজ করে। অ্যাকাডেমির ওই সিনিয়র সিটিজেনদের মহিমাময় হয়ে বিরাজ করার কল্যানেই আমার পাড়ার ছোট ছোট ছেলেদের মধ্যে কম্পিউটার নিয়ে মহিমার ত্রাসের এলিটতার আবহ বানানো হয়, বয়ে নিয়ে যাওয়া হয়। সবকিছুই নিজে করার, নিজে করে মিলিয়ে নেওয়ার উপায় থাকে বলেই, গ্নু-লিনাক্স আসলে এই আবহটার উপরেই আক্রমণ। এই গোটা লেখাটায় যা করে চলার চেষ্টা করছি, পরিশ্রম করো, শেখো, সবকিছু তোমার জন্যে দেওয়া আছে গ্নু-লিনাক্সে। এমনকি আমিও, কম্পিউটারের বাইরের লোক হয়ে, যদি নিজের চেষ্টায় দুবছরেরও কমে এই লেখাটা লেখার মত শিখে উঠতে পারি, তাহলে তুইও পারবি। এটা আমি রোজ অশেষকে পিউকে কেয়াকে বলছি। সম্পদটা সামনেই আছে, দ্যাখ, শেখ। ওই মৌরসি পাট্টার বাইরে, ওদের পাণ্ডিত্যের জনশ্রুতির বাইরে, নিজে হাঁট। নিজে দেখে শেখার চেষ্টা কর, যা যা তোর কাছে আসছে তার সবই এক একটা প্রচার জনশ্রুতি রূপকথা, মূল্যবোধ মনন অবস্থানও এখন একটা জনশ্রুতি, একটা 'নাকি', নিজে হাঁট — যদি তোর ডাক শুনে কেউ না আসে, তবে 'নাকি' একলা চলতে হয় ... (কার্টসি নচিকেতা)

হ্যাঁ, রাইজারএফএস। রাইজারএফএস-এ গোটা তথ্যটাই সজ্জিত থাকে একটা কাঠামোয় যার নাম 'বি-ব্যালান্সড ট্রি'। এই ট্রি বা গাছের আকারে সজ্জার কল্যাণেই খুব চমৎকার ভাবে ডিস্কভূমিটাকে ব্যবহার করতে পারে রাইজারএফএস।

খুব ছোট ছোট ফাইলগুলোকে সরাসরি ওই গাছের পাতায় পাতায় রাখতে রাখতে যায়, শিকড় গুঁড়ি ডালপালা এরকম কোথাও না-রেখে — শুধুমাত্র বাস্তব ডিস্কের বাস্তব শরীরে তার অবস্থানের পয়েন্টারটুকু লিখে রেখে। এই লিখে রাখাটাও এক কেবি বা চার কেবির এক একটা খণ্ডে ঘটেনা, ঘটে ঠিক যতটা আয়তন প্রয়োজন ততটা আয়তনের এক একটা অংশে। রাইজারএফএসের আর একটা ভালো জায়গা আইনোডগুলোকে গতিশীল বা ডায়নামিক রকমে ধার্য করার প্রক্রিয়া। ইএক্সটিটু জাতীয় পুরোনো ফাইলসিস্টেমগুলোর চেয়ে এটা ভালো, যেখানে ফাইলসিস্টেম বানানোর সময়েই স্থির করা থাকে আইনোডদের ঘনত্ব কতটা অব্দি হতে পারে। খুব ছোট ছোট ফাইলের বেলায়, রাইজারে, ফাইল আর ফাইলের ‘stat_data’ বা আইনোড তথ্য পাশাপাশি রাখা থাকে, যাতে মাত্র একবার ডিস্কে পড়ে বা লিখেই ফাইল সংক্রান্ত গোটা কাজটা সিস্টেম করে নিতে পারে। এতে ডিস্ক অনেক দ্রুত কাজ করে। আর জার্নালের কল্যাণে কোনো কেলো বা ক্র্যাশ থেকে ফেরত আসতে রাইজারের সময় লাগে মাত্র কয়েক সেকেন্ড। এমনকি গাবদা গাবদা সাইজের ফাইলসিস্টেমেও।

৩.৪।। জেএফএস

জার্নাল থাকাটা জেএফএস-এর নামেই ঘোষিত — জার্নালিং ফাইল সিস্টেম। এটা এনেছিল আইবিএম। গ্নু-লিনাক্সে জেএফএস এসেছে খুব অল্পদিন, দুহাজার সালে বিটা ভার্শন বা প্রস্তুতিকালীন সংস্করণ, আর ২০০১-এ এসে প্রথম পূর্ণাঙ্গ ভার্শন। মূলত খুব বেশি কাজের চাপে থাকা সার্ভার মেশিনগুলোর কথা ভেবেই জেএফএস তৈরি, কাজ কত দ্রুত করা যায় এটাই যেখানে প্রায় একমাত্র বিবেচ্য। জেএফএস পুরোপুরি একটা ৬৪-বিট ফাইলসিস্টেম, তাই খুব বড় বড় সাইজের ফাইল এবং পার্টিশন বানানো জেএফএসে কোনো সমস্যাই নয়। আমরা এই সেকশনের শেষে বড় ফাইলের আয়তনের কথায় আসছি। সার্ভার সিস্টেমের বেলায় জেএফএস খুবই কার্যকরী।
রাইজারের মত জেএফএস ব্যবস্থাতেও জার্নাল রাখা হয় শুধু মেটাডেটার। খুব বিশদ সর্বব্যাপী চেকিং-এর জায়গায় চেক করা হয় ফাইলসিস্টেমের সাম্প্রতিকতম বদলের কারণে মেটাডেটার শরীরে ঘটে যাওয়া পরিবর্তনগুলোকেই। এতে ক্র্যাশ থেকে প্রত্যাবর্তনের প্রচুর সময় বেঁচে যায়। তথ্য লেখা এবং পড়ার কাজ, এবং তার সঙ্গে সঙ্গে লগ বা নথীর বদলের কাজ একই সঙ্গে একাধিক জায়গায় ঘটতে থাকে জেএফএসে, এক একটা গুচ্ছে বা দলে বা গ্রুপে। আলাদা আলাদা ভাবে লেখার কাজের জায়গায় এই কনকারেন্ট কাজের প্রথা জেএফএসের দ্রুততার একটা বড় কারণ। জেএফএসে দু ধরনের ডিরেক্টরি কাঠামো একই সঙ্গে থাকে। ছোট ছোট ডিরেক্টরির জন্যে ডিরেক্টরির মধ্যেকার মালপত্তর সরাসরি আইনোডে রাখা হয়। আর বড় বড় ডিরেক্টরির বেলায় বি-ট্রি ব্যবহার করা হয়। এতে ডিরেক্টরি সংগঠন খুব ভালো ভাবে হতে পারে। এবং আইনোড ধার্য করার প্রক্রিয়াটা জেএফএস ব্যবস্থাতেও গতিশীল বা ডায়নামিক। আমরা আগেই বলেছি ইএক্সটিটু ফাইলব্যবস্থায় এই আইনোড ঘনত্বটা আগে থেকেই স্থির-করা। তাই, ফাইলসিস্টেমে কতগুলো ডিরেক্টরি থাকতে পারবে, বা ডিরেক্টরিতে কতগুলো ফাইল থাকতে পারবে তার কিছু বিধিনিষেধ এসে পড়ে। জেএফএস ব্যবস্থায় এইসব ঝামেলা নেই, এখানে ডিরেক্টরি বা ফাইলদের ভূমি বণ্টন করা হয় গতিশীল রকমে। আর যেই কোনো একটা ফাইলের বা ডিরেক্টরির কোনো একটা জমির প্রয়োজন ফুরিয়ে যায় অমনি সেই জমিটাকে ফের বণ্টনের জন্যে তুলে রাখে।

৩.৫।। এক্সএফএস

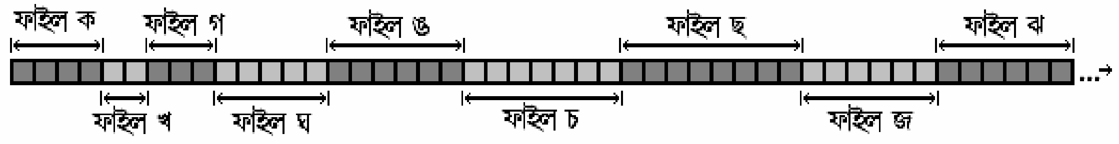
ইউনিক্সের বিভিন্ন ফ্লেভারের কথা বলতে গিয়ে আমরা আইরিক্সের (Irix) কথা উল্লেখ করেছি, চার এবং পাঁচ নম্বর দিনের আলোচনায়। এসজিআই বা সিলিকন গ্রাফিক্স এই এক্সএফএস ফাইলসিস্টেম প্রথম তৈরি করে এই আইরিক্স অপারেটিং সিস্টেমের জন্যে, ১৯৯০-এ। খুব দ্রুত কাজ করতে পারে এমন একটা ৬৪-বিট ফাইলসিস্টেম বানানোর উদ্দেশ্যে। কম্পিউটারের তথা মেশিনের কাছে সিস্টেমের চাহিদা শুধুই বাড়ছিল। বড় বড় ফাইল নিয়ে কাজ করার ব্যাপারে এক্সএফএস খুবই দড়। পরে এক্সএফএসকে গ্নু-লিনাক্সে পোর্ট করা হয়। তবে ঠিক রাইজারএফএস বা জেএফএস-এর মতই এক্সএফএস-ও বেশি মাথা ঘামায় ডেটার চেয়েও মেটাডেটা নিয়ে। উন্নত হার্ডওয়্যারে এক্সএফএস বোধহয় অন্য ফাইলসিস্টেমের চেয়ে দ্রুততর কাজ করতে পারে।
এক্সএফএস ফাইলসিস্টেম তৈরির সময়েই, যে ব্লক-ডিভাইসের উপর ফাইলসিস্টেমটা তৈরি হচ্ছে, সেটাকে আটটা বা তার বেশি সমান মাপের একরৈখিক বা লিনিয়ার অঞ্চলে ভেঙে ফেলা হয়। এদেরকে ডাকা হয় বণ্টন-দল বা

অ্যালোকেশন গ্রুপ বলে। প্রত্যেকটা অ্যালোকেশন গ্রুপের থাকে নিজের নিজের আইনোড এবং মুক্ত ডিস্ক ভূমি। এর মালিক ওই অ্যালোকেশন গ্রুপ। অ্যালোকেশন গ্রুপগুলো হল ফাইলসিস্টেমের মধ্যে ছোট ছোট ছানা ফাইলসিস্টেম। প্রত্যেকটা অ্যালোকেশন গ্রুপ অন্য অ্যালোকেশন গ্রুপদের থেকে স্বতন্ত্র এবং স্বাধীন। তাই কারনেল একই সাথে একাধিক অ্যালোকেশন গ্রুপের সঙ্গে কথোপকথনে আসতে পারে। জেএফএসের কনকারেন্ট ব্যবস্থার কথা মনে করুন। এই অ্যালোকেশন গ্রুপ হল এক্সএফএস ফাইলব্যবস্থার একটা জোরের জায়গা। কিন্তু, স্বাভাবিক ভাবেই, একাধিক প্রসেসর সম্পন্ন কম্পিউটারে, সিমেট্রিক-মাল্টি-প্রসেসিং গোছের, সবচেয়ে ভালোভাবে কাজ করতে পারে এই অ্যালোকেশন গ্রুপের ধারণাটা। এসএমপির কাজের সঙ্গে খুব যায় এই অ্যালোকেশন গ্রুপের ব্যবস্থাটা।
ডিস্কভূমিটাকে নিয়ন্ত্রণের কাজটা এক্সএফএস খুব ভালো ভাবে করে। এক্সএফএসের কাজের দ্রুততার একটা কারণ ওই বি-ট্রি। অন্য ফাইলসিস্টেম থেকে সম্পূর্ণ স্বতন্ত্র এক্সএফএস-এর একটা বৈশিষ্ট্য হল 'ঢিলে বণ্টন' বা ডিলেড অ্যালোকেশন ('delayed allocation')। ডিস্কভূমিতে তথ্য বণ্টনের গোটা কাজটাকে এক্সএফএস দুটো ভাগে ভাগ করে নেয়। স্থগিত রাখা বা দেরি-করিয়ে-দেওয়া বা ডিলেড কাজটাকে রেখে দেওয়া হয় র‍্যামে, আর তার জন্যে চিহ্নিত করে রাখা হয় প্রয়োজনীয় ভূমির পরিমাপ। কিন্তু সত্যিকারের বণ্টনটা করার জন্যে এক্সএফএস একদম শেষ মুহূর্ত অব্দি অপেক্ষা করে। এই ধরে রাখা তথ্যের মধ্যে কোনো অস্থায়ী অংশ যদি থাকে, অনেক তথ্যের চরিত্র তাই হয়, খুব ছোট্ট একটা সময়ের জন্যে জন্মায় তারা, তার পরেই মরে যায়, সেই তথ্য আর কোনোদিনই ডিস্কের গায়ে লেখা হয়না। কারণ, যতক্ষণে এক্সএফএস গোটা তথ্যটাকে লিখে ফেলার সিদ্ধান্ত নেয় তার মধ্যে হয়ত সে মরেই গেছে। একটা জিনিষ খেয়াল করুন, এই প্রক্রিয়ার কারণে মোট লেখালেখির পরিমাণ কিন্তু কমে যাচ্ছে। এটা এক্সএফএস ফাইলব্যবস্থার দ্রুততার একটা কারণ। আর ফাইলব্যবস্থার মধ্যে ফাইলগুলোর যে টুকরো হয়ে যাওয়া বা ফ্র্যাগমেন্টেশনের কথা বলেছিলাম আমরা, সেই ফ্র্যাগমেন্টেশনের পরিমাণও এতে কমে যাবে। অনেক কম সংখ্যক বার তাকে তো বাস্তব হার্ডডিস্কে লিখতে হচ্ছে। কিন্তু এই লেখালেখির প্রক্রিয়ার মাঝপথে সিস্টেমে যদি কোনো ঝাড় নামে, কোনো ক্র্যাশ, তাহলে অবভিয়াসলি তথ্য হারানোর পরিমাণও অন্য ফাইলসিস্টেমের চেয়ে বেশি হবে। এই টুকরো হয়ে যাওয়া বা ফ্র্যাগমেন্টেশন এড়ানোর আর একটা কায়দাও আছে এক্সএফএস-এর। যার নাম প্রাকবণ্টন বা প্রিঅ্যালোকেশন ('preallocation')। ফাইলটাকে ডিস্কের ফাইলসিস্টেমে লেখার আগে এক্সএফএস ফাইলটার সঙ্গে মানানসই কিছুটা ডিস্কভূমিকে রিজার্ভে সরিয়ে রাখে। তাই পরে যখন লেখে, স্বাভাবিকভাবেই, ফাইলটার টুকরো হওয়ার সম্ভাবনা অনেক কমে যায়। আর একটা ফাইলের মোট তথ্য যেহেতু একটা গোটা ডিস্কের নানা জায়গায় ছড়িয়ে থাকেনা, ফাইলসিস্টেমটা কাজও করতে পারে অনেক তাড়াতাড়ি।

৩.৬।। আইএসও-৯৬৬০ (iso9660)
সিডি পোড়ানোর প্রচুর সফটওয়ার গ্নু-লিনাক্সে এমনিতেই দেওয়া থাকে। ছবি-ইঁদুর দিয়েও রাশি রাশি সফটওয়ার আছে। ইন্দ্র জিএলটির মিটিং-এ একটা মজার কথা বলেছিল, এখানে প্রবলেমটা স্কেয়ারসিটি বা সফটওয়ারের অভাব নয়। প্রবলেম অফ প্লেনটি। প্রাচুর্যের সমস্যা। আসলে চারে এবং পাঁচে দেওয়া গ্নু-লিনাক্সের জন্ম এবং বিবর্তনের ইতিহাসটা খেয়াল করুন, তাহলেই বুঝতে পারবেন। ধরুন আপনি প্রোগ্রাম করতে জানেন, আপনি দেখছেন আপনার কোনো একটা কাজ করতে একটা সমস্যা হচ্ছে, আপনি চালু সফটওয়ারটাকে একটু বদলে নিতে চাইলেন। এই কাজের প্রোগ্রামের জন্যে যতগুলো সফটওয়ার পাওয়া যায়, তাদের সবাইকে আপনি নেড়ে ঘেঁটে দেখলেন। গ্নু-লিনাক্স জগতের বাইরে এটা অসম্ভব। কারণ, আপনি তো সোর্স কোড পাবেন না। আপনি প্রোগ্রামটা ব্যবহার করতে পারেন, জানতে বা বুঝতে পারেন না। এবার আপনার প্রয়োজন মোতাবেক সেটাকে মানিয়ে নিতে গিয়ে তৈরি হয়ে গেল একটা আদ্যোপান্ত নতুন প্রোগ্রাম। এবং স্টলম্যানের জিপিএলের খচড়ামিটা মনে করুন, আপনি জিপিএল-এর আওতায় আসা সফটওয়ার ব্যবহার করেছেন মানেই আপনার সফটওয়্যারও জিপিএলে এসে গেল, এটাও একইরকম উন্মুক্ত। এবং আপনি লিনাক্সী মানে আপনিও তো তাই চান। শুধু মাত্র নিজে পয়সা পিটে গাড়ি বাড়ি করলেন, আপনার নিজের পরবর্তী প্রজন্মও জানতে বুঝতে পারলনা, বিদ্যা থেমে গেল, এটা চাইলে তো আপনি গ্নু-লিনাক্সে আসতেনই না। এই প্রোগ্রাম বানাতে অন্য নানা প্রোগ্রামের চমৎকারিত্বটা গল্পের মত করে ভারি চমৎকার পাবেন এরিক রেমন্ডস-এর 'দি ক্যাথিড্রাল অ্যান্ড দি বাজার' লেখাটায় (http://www.tuxedo.org/~esr/)। এভাবেই গ্নু-লিনাক্সে যে কোনো কাজের জন্যে রাশি রাশি প্রোগ্রাম তৈরি হয়ে গেছে। ইন্দ্রও ঠিকই বলেছে, একজন নতুন ব্যবহারকারীর সত্যিই মাথা

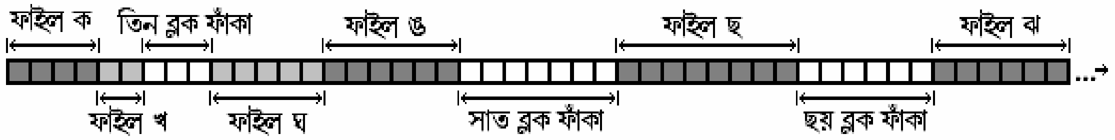
খারাপ হয়ে যায়। তাও তো, এই লেখাটায়, আগেই ঠিক করে নিয়েছিলাম, কোনো গুই প্রোগ্রামকে আনব না। তাই সিডি পোড়ানোর যেসব কমান্ড মোড প্রোগ্রাম দেওয়া থাকে সিস্টেমে, তার মধ্যে দুটো হল সিডিআরডিএও (cdrdao) এবং সিডিরেকর্ড (cdrecord)। দুটোই বেড়ে জিনিষ। এই দুটোই হল, একটা সিডি রয়েছে, বা তার ইমেজ রয়েছে, সেটাকে একটা নতুন সিডিতে পোড়ানোর। কিন্তু ধরুন আপনার কিছু ডিরেক্টরি আর ফাইল আপনি সিডিতে তুলতে চান, সেই অবস্থায় আপনার সেই ডিরেক্টরি এবং ফাইলগুলো থেকে একটা ইমেজ বা প্রতিচিত্র বানাতে হবে। ইমেজটা আসলে কী করবে এখন আপনি সেটার একটা আন্দাজ করতে পারছেন। হার্ডডিস্কে থাকার সময়ে তার যে ফাইলব্যবস্থা ছিল সেটাকে বদলে সিডিতে থাকার ফাইলব্যবস্থায় নিয়ে আসা। এই ইমেজ বানানোর সফটওয়ারও দেওয়া থাকে গ্নু-লিনাক্সের ভিতরেই, কমান্ড মোডে করার, মেকআইএসওএফএস (mkisofs) এবং মেকহাইব্রিডএফএস (mkhybridfs)। নামদুটোকে দেখুন তো ভেঙে বুঝতে পারছেন কিনা। সফটওয়্যার দেওয়ার সঙ্গে সঙ্গে আরো চমৎকার যেটা, সেটা হল, এই প্রোগ্রামগুলোর নিজের ডকুমেন্টেশন, মূলত ম্যানপেজ পড়ে ফেললেই সিডি বিষয়ে বাংলা বাজারে নিজেকে একজন ধুরন্ধর বলে আপনি চালাতে পারবেন। ঠিক যেটা হয় এমপ্লেয়ারের (mplayer) ডকুমেন্টেশন পড়ে। বিভিন্ন ধরনের মাল্টিমিডিয়া ফাইল, তার ফরম্যাট, তার কাজের তফাৎ, একদম মাল্টিমিডিয়ার হদ্দমুদ্দ — যা বেবি, দৌড়ে যাবি, শিখবি মাল্টিমিডিয়া, নমস্তস্যৈ, নমস্তস্যৈ, নমস্তস্যৈ, নমোনমঃ। ম্যানপেজগুলো ব্রাউজ করে একবার দেখে নিন, আইএসও নিয়ে ওরা কী বলেছে।

সিডিরমের ফাইলব্যবস্থা নিয়ে একটু বড় করে এখানে বলতে হচ্ছে, তার কারণ, খেয়াল করুন, যে মাধ্যমটায় এই ফাইলব্যবস্থাটা তৈরি হচ্ছে সেটা একদম আলাদা। আমরা হার্ডডিস্কের চৌম্বক মাধ্যম নিয়ে অনেক কথা বলেছি। সিডির সেই অপ্টিকাল ভৌত মাধ্যমের আলোচনায় আর গেলাম না। শুধু মাধ্যম বদলে যাওয়ায় ফাইলব্যবস্থাটাও তো আমূল বদলে যাবেই, অন্যরকম হয়ে যাবেই। এবং মজার কথা কী বলুন তো, পুরোনো দিনের হার্ডডিস্কেও একসময় এই জাতীয় ফাইলব্যবস্থাই ব্যবহার হত, যা সিডিরমের বেলায় এখন হয়। আমরা আগেও প্রসঙ্গটা এনেছি, পরপর পরস্পর-সন্নিবদ্ধ বা কনটিগুয়াস ফাইলসিস্টেম। এই কনটিগুয়াস ফাইলসিস্টেম কিন্তু একটা জায়গায় ভারি উৎকৃষ্ট জিনিষ, ফাইল টুকরো হয়ে যাওয়ার বা ফ্র্যাগমেন্টেশনের কোনো সমস্যাই তাতে ছিলনা। কিন্তু কাজের ঝামেলার জন্যে সেটা বদলে যায়। পরে যা ফের ফেরত এল সিডি এবং ডিভিডি লেখার ব্যবস্থায়। কারণ এখানে ফাইল বা ডিরেক্টরিগুলো লেখার পরে আর বদলাচ্ছেনা। বারবার পোড়ানোর মত, মানে রিরাইটেবল সিডিতেও ভাবুন, আপনি যে পুড়িয়ে ফেলা ফাইল আর ডিরেক্টরিগুলোকে বদলাচ্ছেন, তা কিন্তু নয়। আপনি পুরোনো পোড়ানোটাকে মুছে ফেলে নতুন করে পোড়াচ্ছেন।



এই ছবিতে দেখুন, একটা কনটিগুয়াস ফাইলব্যবস্থায় ফাইল কী ভাবে থাকে। ফাইলব্যবস্থাটার শুরু থেকে শুরু করে মোট ৪৭-টা ব্লক আমরা দেখিয়েছি। পরপর একরৈখিক বা লিনিয়ার সজ্জায়, ফাইল ক থেকে ফাইল ঝ। ফাইলব্যবস্থাটা কিন্তু এর পরেও চলছে, এখানেই শেষ নয়, আমরা এখানে কয়েকটা মাত্র ফাইলকে দেখালাম। ফাইল ক থেকে ফাইল ঝ পরপর ৪, ২, ৩, ৫, ৬, ৭, ৮, ৬, এবং ৬ ব্লকের। আদতে এরা যে কোনো মাপেরই হতে পারত। ফাইলদের এই পরপর কনটিগুয়াস রকমে থাকাটা বেশ সহজ এবং আরামদায়ক একটা ব্যবস্থা। একটা ফাইলের মোট পরিমাপ মনে রাখা মানে জাস্ট দুটো সংখ্যা মনে রাখা, প্রথম ব্লকের নম্বর, আর ফাইলটায় কটা ব্লক আছে। যেমন ব্লকগুলোর নম্বর যদি ০ থেকে শুরু করি, কম্পিউটার-দুনিয়ায় মোটামুটি সেটাই প্রথা, আমাদের এই পাঠমালাটাও শুরু হয়েছিল ০ থেকে, তাহলে ফাইল ক মানে ০ আর ৪। শুরু ০ নম্বর ব্লক থেকে, চলেছে ৪ ব্লক। ফাইল খ মানে ৫ আর ২। ফাইল গ মানে ৭ আর ৩। ফাইল ঘ মানে ১০ আর ৫। ইত্যাদি। এই ফাইলব্যবস্থার আর একটা সুবিধে এই যে ফাইল পড়ার সময় সিস্টেমের সময় খরচ হবে সবচেয়ে কম, এক একটা ফাইল এক এক বারে গোটাটা পড়ে ফেলছে। একটা ফাইল পড়ার সময় সিস্টেমকে খুঁজতে হবে মাত্র একবার। শুধু শুরুর ব্লকটা গুনে বার করো, মার দিয়া কেল্লা। তার মানে এই কনটিগুয়াস ব্যবস্থাটা একই সঙ্গে সহজ এবং দ্রুত। তাহলে বাতিল হল কেন? গন্ডগোলটা পাকতে থাকে সময়ের সঙ্গে সঙ্গে। আপনি যদি আপনার মেশিনটাকে সচন্দন সবেলপাতা তুলে রাখেন, যেমন

অনেকেরই মেশিন দেখি, ঢাকনা লাগানো থাকে, যেন ওটা কাজের যন্ত্রপাতি নয়, একটা সযত্নে তুলে রাখার গয়না, তাহলে কোনো ঘাপলা নেই। আপনার ফাইলব্যবস্থাটা চিরকাল হুবহু একইরকম চমৎকারী আর মনোহর রয়ে যাবে। কিন্তু যদি আপনি ভুলক্রমেও মেশিনটা ব্যবহার করেন, তাহলেই গেলেন। পরের ছবিটা দেখুন।

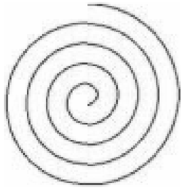


আপনি মেশিনটা ব্যবহার করছেন। ব্যবহার করা মানেই কিছু পুরোনো ফাইল ওড়ানো, কিছু নতুন ফাইল লেখা। ফাইল গ, ফাইল চ এবং ফাইল জ ইহজগতে আর নেই, তাদের আত্মার শান্তিকামনা করে আপনি আরো আরো ফাইল লিখে জীবনে কিছু ফাইল রেখে যাওয়ার দিকে এগোচ্ছেন। এই ফাইল গ চ এবং জ তাদের ডিস্কপাতায় চৌম্বকজলের মত অস্থায়ী জীবন দিয়ে কিছুই যে রেখে যায়নি তা কিন্তু নয়। তারা প্রত্যেকেই ডিস্কের পাতায় এক একটা ফুটো রেখে গেছে, ঠিক যত ব্লকের উপস্থিতি ছিল তাদের, সেই মাপের এক একটা ফুটো। ডিস্কের প্রথম সাতচল্লিশটা ব্লক জুড়ে এখন আছে ছটা ফাইল এবং তিনটে ফুটো। ফুটো তিনটের সাইজ ৩, ৭ এবং ৬ ব্লক। এখনো আপনার সমূহ বেদনার দিন শুরু হয়নি। ফাইলব্যবস্থার আরো পরের আরো পরের দিকে, মানে আমাদের ছবির অদৃশ্য ডানদিকে আপনি ফাইল লিখে এবং লিখে চলেছেন। তাদের কিছু কিছু ওড়াচ্ছেনও নিজের খুশিমত। সেই বাবু আর কোথায় যারা পায়রা ওড়াত এবং বাজি পোড়াত, আমরা শুধু ফাইল ওড়াই এবং সিডি পোড়াই। একসময় আসবে যখন এইভাবে হরেক সাইজের ফুটোয় আর ফাইলে আপনার গোটা ডিস্কটা ভরে যাবে। এবার? এখন সিস্টেমকে ডিস্কে লেখার মত জায়গার হিশেব রাখতে হচ্ছে ফুটো এবং ফুটোর সাইজ গুনে, যাতে সেই ফুটোগুলোকে ঠিক ঠিক সাইজের ফাইলে ভরানো যায়। কোনো ফাইল লেখার আগেই এখন খুঁজে বার করতে হবে সেই ফাইল রাখার জন্যে সঠিক সাইজের একটা ফুটো। আপনি মেজাজে কবিতা লিখতে বসলেন, একটা ওয়ার্ড প্রসেসর খুললেন, প্রথমেই সিস্টেম আপনার কাছে জানতে চাইল, কত সাইজের ফাইল হবে? কারণ, সিস্টেমের তো মাথায় আছে ফুটোময় এই মহাবিশ্ব। আপনি হয়তো, আপনার প্রগাঢ় আত্মকাব্যসচেতনতার বলে বলীয়ান হয়ে, এবং পরে যাতে কাব্যোচ্ছ্বাস বাড়লে ফাইলসাইজের কার্পণ্যে আপনাকে আটকে যেতে না-হয়, কারণ, তখনতো ওই ফুটোয় আর তথ্য রাখা যাবেনা, আপনি জানিয়ে দিলেন, পাঁচ এমবি। সিস্টেম জানাল, বাতিল। ওই সাইজের ফুটো নেই। মাঝখান থেকে আপনার কবিতা লেখার মুডটাই ভেস্তে গেল। আবার ওয়ার্ড প্রসেসর খুলে চার এমবি তিন এমবি করে নিজের উচ্চাকাঙ্খা যে কমিয়ে আনতে পারেন না, তা নয়, কিন্তু, নিজেই ভেবে দেখুন, কবিতা লেখার তথা পিসি ব্যবহারের এটা একটা স্ট্যান্ডার্ড প্রক্রিয়া হল?

কিন্তু এবার ভাবুন, হার্ডডিস্কে যে ব্যবস্থা অসম্ভব, সেটা চমৎকার ভাবে সম্ভব হয়ে ওঠে যখন ফাইল বা ডিরেক্টরির কাঠামো এবং সাইজ এবং সংখ্যা আর বদলাবে না। ঠিক সিডি-রম এবং ডিভিডি-রমের বেলায় যেটা ঘটে। অনেক বছর আগে চৌম্বক হার্ডডিস্কে এই কনটিগুয়াস রকমেই ফাইল রাখা হত, এর সরলতা এবং দ্রুততার জন্যে। কিন্তু সিডি-রম, ডিভিডি-রম এই জাতীয় অপ্টিকাল মাধ্যমে আবার ফেরত এসেছে, যেখানে ফাইলব্যবস্থা মানে একবার লেখো এবং বারবার পড়ো। আইএসও-৯৬৬০ হল সিডি-রমের ফাইলব্যবস্থা। রম মানে মনে আছে তো, রিড-অনলি, শুধু পড়া যায়। এই রিড-অনলি ফাইলসিস্টেমের একটা আরাম ভাবুন, ফ্রি-ব্লক কী রইল তা নিয়ে মাথা ঘামাতে হয়না, আর তো লিখতে হবেনা। রিরাইটেবল সিডি মানে যেখানে পুরোনো তথ্য মুছে নতুন করে লেখা যায়, বা মাল্টিসেসন সিডি, মানে যেখানে আপনি সিডির গোটা তথ্যটা এক বারের জায়গায় একাধিক বারে পোড়াচ্ছেন, সেখানেও, লেখার মুহূর্তে গোটা ব্যবস্থাটা কিন্তু প্রদত্ত, পড়ার সময়ে তারা আর বদলাচ্ছে না। 'ইন-সিডি' জাতীয় যে ব্যবস্থাগুলো আছে, সিডিকেই ফ্লপির মত করে ব্যবহার করার, তাদের ফাইলব্যবস্থাগুলো, স্বাভাবিক ভাবেই, আলাদা।

আইএসও-৯৬৬০ হল সবচেয়ে জনপ্রিয় এবং বহুল ব্যবহারের সিডি-রম ফাইলব্যবস্থা। আইএসও-৯৬৬০ হল একটা আন্তর্জাতিক স্ট্যান্ডার্ড, অষ্টাশিতে গৃহীত হয়েছিল। বাজারে প্রাপ্তব্য প্রতিটি সিডি-রমই প্রাথমিকভাবে এই স্ট্যান্ডার্ড মেনে চলে, তার সঙ্গে কোথাও কোথাও আরো বাড়তি কিছু থাকে। এই স্ট্যান্ডার্ড গৃহীত হয়েছিল যাতে এই গ্রহের প্রতিটি সিডি-ড্রাইভেই যে কোনো সিডি-রম পড়া যায়, যে অপারেটিং সিস্টেমেই সেটা লাগানো হোক, যে রকম বাইট-ব্যবস্থাই ব্যবহার করা হোক সেই সিস্টেমে। পোসিক্স বা এলএসবি স্ট্যান্ডার্ডের ঠিক যে যুক্তি, আগেই বলেছি।

হার্ডডিস্কের বেলায় আমরা যে এককেন্দ্রিক সিলিন্ডারের কথা বলেছি, সিডি-রমে ওরকম কিছু থাকে না। তার বদলে থাকে একটা ছেদহীন স্পাইরাল বরাবর পরপর সাজানো বিটের সারি। একরৈখিক বা লিনিয়ার। তবে খোঁজার সময় স্পাইরালটার শুরু থেকেই শুরু করতে হবে, তা নয়। যে কোনো জায়গাতেই খোঁজা যেতে পারে। স্পাইরাল শব্দটার কোনো বাংলা প্রতিশব্দ আছে বলে জানিনা। একটা ছবি দিই। একটা বৃত্তাকার পথ, যার পরিধি ক্রমে কমছে বা বাড়ছে, যে ভাবেই ভাবুন। আপনাদের জন্যে আমি একটা স্পাইরাল আঁকার চেষ্টা করলাম, দু-ঘন্টা ধরে। পারলাম না। এটা শেষে নেটে গিয়ে নামিয়ে একটু বদলে নিলাম কপিরাইট বাঁচাতে। কিন্তু এরাই বা আঁকল কী করে? যন্ত্র দিয়ে মনে হয়। বা হাতে এঁকে স্ক্যান করে? কে জানে বাপু? প্রকৃতি ভীষণ স্পাইরালপরায়ণ। একটা শাঁখের মুখটা ভাবুন। স্পাইরাল আর ফিবোনাক্কি সংখ্যার সিরিজ নিয়ে ভারি মজা আছে। ফিবোনাক্কি সিরিজ মানে ০, ১, ১, ২, ৩, ৫, ৮, ১৩, ২১ ... এই সংখ্যার সারিটা, যার প্রত্যেকটাকেই পাওয়া যায় আগের দুটোকে যোগ করে। টিভিতে একদিন দেখাচ্ছিল, প্রকৃতির সব স্পাইরালগুলো এই ফিবোনাক্কি সিরিজ মেনে চলে।



যাকগে, সিডিরমের কথায় আসা যাক। সিডিরমের এই স্পাইরাল বরাবর বাইটগুলো পরপর সাজানো থাকে লজিকাল ব্লকের এককে। যার এক একটার মাপ ২৩৫২ বাইট। এর মধ্যে অনেকটা থাকে প্রিঅ্যাম্বল মানে গৌরচন্দ্রিকা — ধরুন মেটাডেটা গোছের, কী তথ্য কী ভাবে আছে সেই বিষয়ে তথ্য। এর সঙ্গে এরর কারেকশন বা ত্রুটি সংশোধনও থাকে। মানে তথ্যের আকারটা সঠিক আছে কিনা সেটা মিলিয়ে নেওয়ার কৌশল। আর জেনুইন তথ্য থাকে ২০৪৮ বাইট। গানের অডিয়ো সিডি থাকলে গানের আগের, পরের, মধ্যের নৈঃশব্দ্য। যা ডেটা সিডি বা তথ্যের সিডিতে থাকেনা। সিডি পোড়ানোর সময় দেখবেন, অনেক সময় মিনিট আর সেকেন্ডের মাপে গোটাটা বলছে, এই মাপটা হল ১ সেকেন্ড = ৭৫ ব্লক। প্রত্যেকটা সিডিরমের গোড়ায় থাকে ফাঁকা ষোলোটা ব্লক। ফাঁকা কারণ এগুলো নিয়ে আইএসও স্ট্যান্ডার্ডে কিছু বলে দেওয়া হয়নি। বুটেবল সিডি বানানোর জন্যে, মানে যা দিয়ে বুট করা যায়, অপারেটিং সিস্টেম বা অপারেটিং সিস্টেম চালু করে দেওয়ার যন্ত্রপাতি যাতে দেওয়া থাকে — এই বুট সিডি বানানোর জন্যে এই ষোলোটা ব্লক ব্যবহার করা হয়। অন্য কাজেও কেউ ব্যবহার করতে পারে। এরপর আসে প্রাইমারি ভলিউম ডেসক্রিপটর। এখানে একটা সিডিকে একটা সেটের একটা ভলিউম হিশেবে ভাবুন। এবার এই প্রাইমারি ভলিউম ডেসক্রিপটরে থাকে ওই সেটটার পরিচিতি, ভলিউমটার পরিচিতি, প্রকাশক এবং তথ্যটা যে সিডির জন্যে প্রস্তুত করেছে, তার পরিচিতি। সব সিস্টেমেই সিডিটা যাতে সমানভাবে পাঠযোগ্য হয়, তার জন্যে প্রাইমারি ভলিউম ডেসক্রিপটরের বিবরণে ব্যবহার করতে হবে শুধু বড় হাতের বর্ণমালা, সংখ্যা, আর অল্প কয়েকটা নির্দিষ্ট যতিচিহ্ন।

সিডিরমটার রুট ডিরেক্টরি এবং অন্য ডিরেক্টরিগুলো ভরা থাকে তাদের মধ্যেকার সাবডিরেক্টরি এবং ফাইলগুলোর এন্ট্রি দিয়ে। খেয়াল রাখুন, এটাও সেই অর্থে মেটাডেটা। এর শেষ ডিরেক্টরির শেষ এন্ট্রির মাথায় থাকে একটা বিশেষ বিট যা দিয়ে বোঝা যায় যে এখানে এই ডিরেক্টরি এন্ট্রিদের বিবরণ শেষ হল। ডিরেক্টরি এন্ট্রিগুলোয় থাকে ফাইলের ঠিকানা, ফাইলের সাইজ, ফাইলের তারিখ আর সময়, ফাইলের নাম। ফাইলের ঠিকানা মানে তার শুরুর ব্লকের নম্বর। ফাইলের নামের একটা কাঠামো আছে। মূল নাম বা বেসনেম, তারপরে একটা '.', তারপরে এক্সটেনশন, তারপরে একটা সেমিকোলন বা ';', এবং সবার শেষে একটা বাইনারি ভার্শন নাম্বার। বেসনেম এবং এক্সটেনশন অংশে ছোটহাতের বা বড়হাতের বর্ণমালা এবং শূন্য থেকে নয় অব্দি অঙ্ক ব্যবহার করা যাবে। মূল নাম বা বেসনেমে আটটা অব্দি বর্ণ ব্যবহার করা যাবে, এবং এক্সটেনশনে তিন অব্দি। আপনার কি একটুও চেনা লাগছে এই হিশেবটা? ছয় নম্বর দিনের আলোচনার সঙ্গে মিলিয়ে?

এই তারিখ আর সময়ের ব্যাপারটায় একটু মজা আছে। আমরা Y2K বা ২০০০ সালে কম্পিউটার সিস্টেমের সময় এবং হিশেব রাখার সমস্যা নিয়ে বহ্বারম্ভে লঘুক্রিয়ার গল্প সবাই জানি। ঠিক এরকমই একটা সমস্যা আছে সিডিতেও। এই সময়ের ফিল্ডটায় একটা আলাদা বাইট করে দেওয়া থাকে বছর, মাস, দিন, ঘন্টা, মিনিট, সেকেন্ড, আর টাইমজোনের প্রত্যেকটার জন্যে। টাইমজোন বা সময়এলাকা ভৌগোলিক অবস্থান, গ্রিনউইচের তুলনায়, যেভাবে আমরা একটা দেশের সময় থেকে আর একটা দেশের সময়কে আলাদা করি। এবার, মজাটা ওই বছরের বাইট-টাতে। সময় গোনা শুরু হয়েছিল, ওই আইএসও কমিটির সিদ্ধান্ত মোতাবেক ১৯০০ থেকে। এরপর ২৫৬ বছর অব্দি, মানে ২১৫৫ সাল অব্দি সিডিতে বছরসংখ্যা লেখা যাবে। তার পরে আবার এটা ফিরে যাবে ১৯০০-তে। তবে কোনো ভয়

নেই, ২১৫৫ অব্দি কেন ২০০৫-অব্দিই সিডি থাকে কিনা সন্দেহ। পশ্চিমে সিডি এখন বাতিল প্রকৌশলের দলে। এখানেও, আমাদের লাগেরই, যে কজনেরই একটু পয়সা আছে, সব ডিভিডি এবং ডিভিডি ড্রাইভ ব্যবহার করছে। ডিভিডি বার্নারের এখনো অবশ্য অশ্লীল দাম। আচ্ছা, বলুন তো, কেন ২১৫৫ সালের পরে গিয়েই সমস্যাটা হচ্ছে? যদি ভেবে উঠতে না-পারেন, আপনার সময় এসে গেছে শূন্য থেকে শুরু করার, মানে, ফের শূন্য নম্বর দিন থেকে একবার পড়তে পড়তে আসুন।

ডিরেক্টরি এন্ট্রিগুলো রাখা থাকে বর্ণানুক্রমে, পরপর অক্ষর অনুযায়ী। শুধু প্রথম দুটো এন্ট্রি বাদ দিয়ে। প্রথমটা হল ডিরেক্টরিটা নিজেই। ঠিক গ্নু-লিনাক্সে যেমন, '.' মানে সেই ডিরেক্টরিটা নিজেই, আর '..' মানে জনিতৃ ডিরেক্টরিটা, মানে, যে ডিরেক্টরির সাবডিরেক্টরি এই ডিরেক্টরিটা। ডিরেক্টরি এন্ট্রিগুলোর প্রথম দুটো সবসময়েই থাকে এই দুটো এন্ট্রি। একটা ডিরেক্টরিতে কতগুলো এন্ট্রি থাকতে পারবে তার কোনো নিষেধ নেই, কিন্তু কটা স্তর অব্দি নামা যাবে, তার নিয়ম আছে। এই ডিরেক্টরির মধ্যে সাবডিরেক্টরি, মা থেকে ছানা, এই ভাবে নামা যাবে আটটা স্তর অব্দি, একে বলে নেস্টিং (nesting)। ডিরেক্টরি এন্ট্রির একদম শেষে এসে আর একটা জিনিষ থাকতে পারে, তার নাম 'সিস্টেম ইউজ' (System Use)। এই ফিল্ডটা সবসময় সব সিস্টেমে থাকেনা, এবং অনেক সিস্টেম এদের আলাদা আলাদা রকমে ব্যবহার করে। এই নেস্টিং আর ফাইলনামের কাঠামোর বিধিনিষেধ আর 'সিস্টেম-ইউজ' নামে ফিল্ডটার কথায় আসছি একটু বাদেই। সিডিরমের শুরুতেই এই ডিরেক্টরি এন্ট্রিগুলো শেষ হলে শুরু হয় ফাইল। ফাইলগুলো রাখা থাকে ওই পরপর কনটিগুয়াস ব্লকে। যেমন আগেই বলেছি, ডিরেক্টরি এন্ট্রিতে দেওয়া ফাইলের শুরুর ব্লকের ঠিকানা আর ফাইলের সাইজ দিয়ে একটা ফাইলকে পুরোপুরি বুঝে নেওয়া যায়। খুব ছোট করে, খুব সরল করে এই হল আইএসও ৯৬৬০ ফাইলব্যবস্থার কাঠামো।

> শুনুন, একদম বাইরের একটা মজা আপনাদের না-শুনিয়ে পারছিনা, এইমাত্র, এই শৈত্যতরঙ্গাক্রান্ত সকালে, একটা অটো ঢুকল আমাদের পাড়ায়। প্রত্যেকটা মফস্বলের কিছু নিজস্ব ভাঁড়ামি থাকে, মহানগরের ভাঁড়ামি থেকে স্বতন্ত্র — সেরকমই একটা, পুষ্পপ্রদর্শনী, মধ্যমগ্রামে হর বছর হয়। ওসব ডেসিবেল ছাড়ুন, মাইক নিয়ে প্রবল চেঁচাচ্ছে। একবার ভাবলাম বাইরে গিয়ে জিগেশ করি, পুষ্পকে যে প্রদর্শন করবেন, পুষ্পর বরের কোনো আপত্তি নেই তো? এর মধ্যেই এল ক্লাইম্যাক্সটা। এরা চেঁচানো ছেড়ে গান বাজানো শুরু করল, অটোটা পাড়া পেরিয়ে যেতে যেতে গানটার বেশ কয়েকলাইন শোনা গেল। কী গান হতে পারে, একবার মনে মনে কল্পনা করুন, যদি পারেন, এইমাত্র শীতের সকালে আমায় যে গরম মাছভাজাটা দেওয়া হল, কাল আমার জলের ফিল্টার পরিস্কার করার বখশিশ হিশেবে, সেটা আপনাকে দিয়ে দেব। গানটা কী, পারলেন ভাবতে? দেবব্রতর 'পুষ্প দিয়ে মারো যারে —'। পাগলা মাছ খাবি তো মাটি খা।

আইএসও ৯৬৬০-র তিনটে পর্যায় বা লেভেল আছে। লেভেল ওয়ানে বিধিনিষেধ খুব বেশি। ফাইলনামের ওই ৮+৩ কাঠামোটা রাখতেই হবে, বর্ণমালার নিয়মও মানতে হবে, ফাইলদেরও একদম নিখুঁতভাবে পরপর সন্নিহিত বা কনটিগুয়াস হতে হবে। এছাড়া আরো আছে। ডিরেক্টরি নামের কোনো এক্সটেনশন থাকবে না, তাদের হতে হবে আট অক্ষরের মধ্যে। এটা এসেছিল, বুঝতে পারছেন, এমএস-ডস ফাইলসিস্টেমের সঙ্গে মিলিয়ে। দু নম্বর লেভেলে নামের দৈর্ঘের নিষেধটা একটু শিথিল। সেখানে নামের অক্ষরের সংখ্যা হতে পারে ৩১ অব্দি, কিন্তু সেই বর্ণগুলো আসবে নিয়ম মেনে। তিন নম্বর লেভেলে ফাইলগুলোকে আর নিখুঁত ভাবে কনটিগুয়াস না-হলেও চলে, একটা ফাইল একাধিক অংশে ভাগ করা থাকতে পারে, তার এক একটা অংশের ব্লকগুলো কনটিগুয়াস হলেই হবে। এই রকম আরো কিছু।

আইএসও ৯৬৬০ প্রথার এই বিধিনিষেধগুলোকে বদলাতে এল রকরিজ এক্সটেনশন (Rock Ridge Extension)। রকরিজ নামটা এসেছে জিন ওয়াইল্ডারের 'ব্লেজিং স্যাডলস' নামে একটা সিনেমার একটা শহরের নাম থেকে, কারণ, যে কমিটি এই প্রথাটা স্থির করে তাদের একজনের খুব পছন্দ ছিল সিনেমাটা। রকরিজ এক্সটেনশন দিয়ে ফাইলের মালিকানা এবং অনুমতির যে কাঠামোটার আলোচনা করেছি আমরা সেটা রাখা চলে একটা সিডিরমে। রাখা যায় মেজর এবং মাইনর ডিভাইস নম্বর আর সিম্বলিক লিংক। ফাইলনামের দৈর্ঘ এবং বর্ণমালা সংক্রান্ত কোনো বিধিনিষেধ থাকেনা। রকরিজ দিয়ে আরো কিছু করা যায় — তার মধ্যে সবচেয়ে জরুরি পুনঃস্থাপন বা 'রিলোকেশন' (relocation)। যে রিলোকেশন দিয়ে ওই আট স্তর অব্দি নেস্টিং-এর নিষেধটাকে এড়ানো যায়। একটা ডিরেক্টরিকে আপনি রিলোকেশন করলেন এবং তার মধ্যে আবার আটটা স্তর অব্দি নেমে গেলেন। এই ভাবে। যে সিস্টেম রকরিজ চেনেনা সে এই সিডিরমটাকে রকরিজ বাদ দিয়ে এমনি সিডিরমের মত করেই পড়ে। আপনি কি এই রকরিজ

এক্সটেনশন কারা এনেছে সেটা নিজে বুঝে নিতে পারছেন? আইএসও ৯৬৬০ আসার পরপরই ইউনিক্সের বিভিন্ন ঘরানার লোকেরা মিলে যখন রকরিজ এক্সটেনশন প্রথা চালু করে তখনো গ্নু-লিনাক্সের জন্ম হয়নি। একটা জিনিষ খেয়াল করুন, রকরিজ যে বিধিনিষেধগুলোকে ভাঙল তার মধ্যে নামের দৈর্ঘের ফ্যাচাংটা কিন্তু উইনডোজ ৯৫ এবং তার পরবর্তী উইনডোজ ব্যবস্থার বেলাতেও অসুবিধে। কারণ, সেখানে নামের দৈর্ঘের কোনো নিষেধ নেই। এইজন্যে এল রকরিজ ছাড়াও আরো একটা জিনিষ, তার নাম জলিয়েট এক্সটেনশন (Joliet Extension)। জলিয়েট প্রথায় লম্বা ফাইলনাম আনা গেল, বর্ণমালার বিধিনিষেধ প্রায় নেই হয়ে গেল, আটটার বেশি নেস্টিং করা গেল, এবং সবচেয়ে মজার কথা ডিরেক্টরি নামেরও এক্সটেনশন দেওয়া গেল। এই শেষটা খুব অদ্ভুত। কারণ, উইনডোজে ডিরেক্টরি নামের এক্সটেনশন ব্যবহার হয়না।

আপনার ম্যানপেজ পড়ার একটা টাস্ক দেওয়া যাক। হার্ডডিস্কে থাকা কোনো একটা ডিরেক্টরি আর ফাইলের সমাহারকে সিডিরমের ফাইলসিস্টেম বানানোর সময়, আগেই বলেছি, একটা উপায় হল 'mkisofs'। এবার, ধরুন আমাদের জিএলটির সিডির জিনিষপত্তরগুলো, কয়েকহাজার ওয়েবপেজ আর কয়েকশো পিডিএফ সহ, রাখা আছে আমার মেশিনের '/arkive' ডিরেক্টরির মধ্যে '/arkive/linux.books' নামে একটা সাবডিরেক্টরিতে। এই গোটা ডিরেক্টরিটাকে আমি একটা ইমেজ করতে চাইছি 'glt-iso' নামে। এই ইমেজটাকে বানানোর জন্যে এর জন্যে সচরাচর আমি কমান্ড দিই, '`mkisofs -DJU -joliet-long -o glt-iso /arkive/linux.books`'। আপনি ম্যানপেজ পড়ে দেখুন তো, এই বাড়তি অপশানগুলো আমি কেন দিই? একটা কথা বলে রাখি, নামের বা নেস্টিং-এর কোনো বিধিনিষেধ মানা হয়নি, এবং আমি চাই আমাদের জিএলটির সিডিটাকে গ্নু-লিনাক্স বা উইনডোজ যে কোনো অপারেটিং সিস্টেম থেকেই সমানভাবে পড়া যাক। এর পরে, ইমেজ বানানো যেই হয়ে গেল, সিডিতে পোড়াতে হয় ইমেজটা। তখন ব্যবহার করছি 'cdrecord'। এখানেই বা অপশানগুলোর মানে কী? শুধু একটা জিনিষ বলি, সিডিরেকর্ড ব্যবহার করার আগে রুট হয়ে নেবেন। আর আপনার ডিভাইস কোনটা জানার জন্যে আপনি ব্যবহার করবেন 'cdrecord scanbus', এটাও রুট হয়ে। এটা আপনি ম্যান থেকেই পেতে পারতেন, কিন্তু কী করব বলুন, আমার মনটা এত নরম। এবার বলুন তো, '`-V "glt-mad@ilug-cal.org"`' অপশানটা 'mkisofs'-এর কমান্ডটার মধ্যে যোগ করলে তার মানে কী হত?

৩.৭।। দুম্বো সাইজের ফাইল আর ফাইলব্যবস্থা

এই গেল ছটা ফাইলসিস্টেম বুড়ি ছুঁয়ে আসা। এর পরেও আরো বহু ফাইলসিস্টেম ব্যবহার করা যায় গ্নু-লিনাক্সে। কিন্তু সেগুলো অনেক বিশেষ ধরনের প্রয়োজন। এখানে উল্লেখ করার মানে পাচ্ছি না। আর যদি তেমন দরকার পড়ে আপনি তো কোলকাতা লাগ এর মেইলিং লিস্টে (ilug-cal@ilug-cal.org) লিখতেই পারেন। মনে করে দেখুন, '`cat /proc/filesystems`' করে আমরা যে ফাইলসিস্টেমগুলোর তালিকা আমরা পেয়েছিলাম, তার বেশ কয়েকটার আলোচনা হল, কিন্তু কতকগুলোর হয়নি। সেগুলোর বিষয়ে প্রাথমিক ধারণা আপনার সিস্টেমের ভিতরেই কোথায় পাওয়া যাবে বলুন তো?

আগেই বলেছিলাম এই সেকশনের শেষে আমরা আসব একটা ফাইলসিস্টেম কত বড় সাইজের অব্দি হতে পারে, আর একটা ফাইলসিস্টেমে কোনো একটা একক ফাইল কত বড় সাইজের অব্দি হতে পারে, তার হিশেবে। এই হিশেবটা কিন্তু রোজই বদলাচ্ছে। রোজই নতুন প্রকৌশল আসছে, পুরোনো প্রকৌশল বদলাচ্ছে, সফটওয়ার এবং হার্ডওয়ার দুটো স্তরেই। তবু, একটা আন্দাজ করা যাবে। বড় ফাইলব্যবস্থার এবং ফাইলের যে বিকট হিশেব এখানে এসেছে, সেই সাইজের কোনো কিছু আমি দেখা বা চেনা তো দূরের কথা, ভেবেও উঠতে পারিনা। এখানে আমি যে হিশেবটা দিচ্ছি সেটা সুজে সাইট থেকে পাওয়া, এবং তাও একদম নতুন না।

হিশেবটা দেওয়ার আগে, বিট বাইট এবং তাদের বৃহত্তর এককগুলো নিয়ে দু-একটা কথা বলে নিই। শূন্যয় বলেছি আমরা, পরেও এসেছে, কিন্তু তখন আমরা বিট বাইটের উপর কিলো মেগা গিগা নিয়েই খুশি ছিলাম, টেরা-পেটা-এক্সা-জিটা-ইয়োটার বিকটতায় পৌঁছইনি। আর, এখানে এই পরিমাপগুলোর স্ট্যান্ডার্ড নিয়েও কিছু কথা বলার আছে। আমরা একাধিকবার বলেছি, কম্পিউটারের তথ্যের পরিমাপটা হয় বাইনারি পদ্ধতিতে, কিন্তু মজার কথা হল, এর গুণিতক বা মাল্টিপলগুলোর জন্যে যে প্রাকশব্দ বা প্রিফিক্স আমরা ব্যবহার করি, যেমন, কিলো, সেটা কিন্তু মেট্রিক পদ্ধতির। কিলোবাইট মানে আমরা ধরছি ১০২৪ বাইট, অথচ, মেট্রিক ব্যবস্থায় কিলো হল ১০০০। কারণ মেট্রিক

পদ্ধতি হল ডেসিমাল বা দশমিক। এখানে, আত্মীয়তাটা এইরকম যে ডেসিমাল সংখ্যা ১০০০ বা ১০^৩-এর সবচেয়ে কাছে বাইনারি প্রথায় পৌছচ্ছি ১০২৪ বা ২^১০। ('^' দিয়ে আমরা পাওয়ার বা সূচক বুঝিয়েছি, ২^৩ মানে ২×২×২) তাই মেট্রিক কিলো শব্দটা বাইনারিতে আসছে যখন, ১০০০ থেকে বদলে ১০২৪ হয়ে যাচ্ছে। আবার কিলো বলতে দশমিক কিলোও চালু ছিল, তখন লেখা হত ১ কিলোবাইট (বাইনারি) মানে ১০২৪ বাইট, ১ কিলোবাইট (দশমিক) মানে ১০০০ বাইট। কিন্তু এতে ভজকটোটা বেড়েছে বই কমেনি। '৯৮ সালের ডিসেম্বরে আইইসি (IEC — **I**nternational-**E**lectrotechnical-**C**ommission) একটা নতুন আইইসি স্ট্যান্ডার্ড চালু করে। তাতে বলা হয়, মেট্রিক ওই প্রাকশব্দ বা প্রিফিক্স বদলে বাইনারি প্রক্রিয়ার গুণিতকগুলোর জন্যে নতুন কিছু প্রাকশব্দ ব্যবহার করতে হবে। এই নতুন প্রাকশব্দগুলো তৈরি হবে মেট্রিক প্রাকশব্দের প্রথম দুই বর্ণ আর 'বাইনারি' এই শব্দটার প্রথম দুই বর্ণ মিলিয়ে। যেমন কিলোবাইট শব্দটা এখন থেকে হবে কিবিবাইট। কিবি (Kibi) মানে, কিলো (Kilo) থেকে আসছে কি (Ki) আর বাইনারি (Binary) থেকে আসছে বি (bi)। মেগাবাইটের 'মেগা' বদলে ব্যবহার করতে হবে মেবিবাইট, গিগাবাইটের 'গিগা' বদলে ব্যবহার করতে হবে গিবিবাইট, ইত্যাদি। কিন্তু এখনো আইইসি ব্যবস্থা তেমন জনপ্রিয় হয়নি। আমরা তাই সাবেক বাইনারি রকমেই হিশেবটা দিচ্ছি।

এক বিট (bit) = ০ বা ১, একক লেখা হয় 'b' দিয়ে।
এক বাইট (byte) = ৮ বিট, একক লেখা হয় 'B' দিয়ে।
এক কিলোবাইট (Kilobyte) = ১০২৪ বাইট = ২^১০ বাইট, একক লেখা হয় 'KB' দিয়ে।
এক মেগাবাইট (Megabyte) = ১০২৪ কিলোবাইট = ২^২০ বাইট, একক লেখা হয় 'MB' দিয়ে।
এক গিগাবাইট (Gigabyte) = ১০২৪ মেগাবাইট = ২^৩০ বাইট, একক লেখা হয় 'GB' দিয়ে।
এক টেরাবাইট (Terabyte) = ১০২৪ গিগাবাইট ২^৪০ বাইট, একক লেখা হয় 'TB' দিয়ে।
এক পেটাবাইট (Petabyte) = ১০২৪ টেরাবাইট = ২^৫০ বাইট, একক লেখা হয় 'PB' দিয়ে।
এক এক্সাবাইট (Exabyte) = ১০২৪ পেটাবাইট = ২^৬০ বাইট, একক লেখা হয় 'EB' দিয়ে।
এক জিটাবাইট (Zettabyte) = ১০২৪ এক্সাবাইট = ২^৭০ বাইট, একক লেখা হয় 'ZB' দিয়ে।
এক ইয়োটাবাইট (Yottabyte) = ১০২৪ জিটাবাইট = ২^৮০ বাইট, একক লেখা হয় 'YB' দিয়ে।

এবার এতাবৎ মহতী এককের বলে বলীয়ান হয়ে আসুন আমরা সেইসব জগদ্দল ফাইলব্যবস্থা আর ফাইল নিয়ে নাড়াচাড়া করি, ভেজানো মুগ আর বাদাম রেখেছেন তো? একদম প্রথমে গ্নু-লিনাক্সে দুই জিবি অব্দি সাইজের ফাইল বানানোর সংস্থান ছিল। কিন্তু তখনো মাল্টিমিডিয়া সেভাবে সিনে আসেনি। আর খুব বড় বড় সাইজের ডেটাবেস বা তথ্যভাণ্ডার নিয়ে কাজ করতে হয়নি। পরপর বড় বড় সার্ভারে যখন গ্নু-লিনাক্স দিয়ে কাজ হতে শুরু করল, কারনেল আর সি লাইব্রেরিগুলোকেও সেইমত বদলে নেওয়া হল যাতে দুই জিবির চেয়ে বড় সাইজের ফাইলও বানানো এবং ব্যবহার করা যায়। এককথায় এর নাম এলএফএস (LFS — **L**arge-**F**ile-**S**upport)। এখন প্রায় সবগুলো ফাইলব্যবস্থাতেই এলএফএস চালু আছে, যাতে খুব উন্নত ধরনের কাজকর্ম কম্পিউটারে করা যায়।

ফাইলসিস্টেম	ফাইল-সাইজের সীমা	ফাইলসিস্টেম-সাইজের সীমা
ইএক্সটিটু বা থ্রি, ১ কেবি ব্লক	১৬ জিবি (GB)	২ টিবি (TB)
ইএক্সটিটু বা থ্রি, ৪ কেবি ব্লক	২ টিবি (TB)	১৬ টিবি (TB)
রাইজারএফএস ৩.৫	৪ জিবি (GB)	১৬ টিবি (TB)
এক্সএফএস	৮ ইবি (EB)	৮ ইবি (EB)
জেএফএস, চার কেবি ব্লক	৮ ইবি (EB)	৪ পিবি (PB)

এর বাইরে গ্নু-লিনাক্স কারনেলের কিছু আভ্যন্তরীন ফাইলসাইজ সীমা আছে। ৩২-বিট সিস্টেমে একটা ফাইল বা ব্লক ডিভাইসের সর্বোচ্চ সাইজ হতে পারে ২ টিবি (TB)। তবে লজিকাল এলভিএম (LVM — **L**ogical-**V**olume-**M**anager) ব্যবহার করে এর চেয়েও বড় ফাইলব্যবস্থা বানানো যায়। ৬৪-বিট সিস্টেমে এই সীমাটা হল ৮ ইবি (EB)। উপরে আমরা কয়েকটা মাত্র ফাইলসিস্টেমের উদাহরণ দিলাম, এরকম সব ফাইলসিস্টেমেই দেওয়া যায়। এসব আমাদের বাড়ির নিজের পিসিতে কার লাগে কে জানে? আমার মেশিনে সবচেয়ে বড় ফাইল বানিয়েছিলাম, টেন কমান্ডমেন্টস এর চারটে ভিসিডি থেকে একটা এভিআই (*.avi) ফাইল বানিয়েছিলাম, এমপ্লেয়ারে ক্যাট করে করা

যায়। তার সাইজ ছিল আড়াই জিবি মানে ২.৬ গিগাবাইট। এটাই আমার টাইটানমানসের চরম। তথাগত ডিভিডি এনকোড ডিকোড করে, ওর হয়ত চার পাঁচ জিবি অব্দি লাগতে পারে। এগুলো সব তাদের লাগে যাদের সঙ্গে কম্পিউটারের সম্পর্ক আমার মত অটেকনিকাল নয়।

৪।। তুমি যে আমার, মাউন্ট, তুমি যে আমার

'মাউন্ট করা' — এই কথাটার হুবহু সংজ্ঞাটা হল একটা পার্টিশনে থাকা একটা ফাইলসিস্টেম (তথ্য-কাঠামো তথা ফাইল রাখার ব্যবস্থা অর্থে, যেমন এক্সএফএস বা ইএক্সটিথ্রি, আবার ডিরেক্টরিব্যবস্থা অর্থেও) একটা ঐক্যবদ্ধ ডিরেক্টরি হায়েরার্কিন্যস্ত ফাইলসিস্টেমে যুক্ত হয়ে পড়া। ঐ পার্টিশনের ফাইলব্যবস্থার মধ্যেও একটা হায়েরার্কিন্যস্ত ডিরেক্টরি ব্যবস্থা আছে। পার্টিশনের ফাইলব্যবস্থাটুকুর একটা রুট ডিরেক্টরি আছে। সেই রুট ডিরেক্টরি থেকে শাখা-প্রশাখার মত বেরিয়ে এসেছে নানা ডিরেক্টরি সাবডিরেক্টরি এবং তাদের মধ্যে থাকা ফাইল, যেমন আমরা দেখিয়েছি সাত নম্বর দিনের ১১ নম্বর সেকশনের ছবিতে। ঐক্যবদ্ধ মূল ফাইলসিস্টেমের তেমনি আবার একটা রুট আছে, এবং সেখান থেকে বেরোনো শাখাপ্রশাখা। পার্টিশনের ওই ফাইলব্যবস্থার রুট ডিরেক্টরি এখন মূল রুট ডিরেক্টরির মধ্যের কোনো একটা ডিরেক্টরি বা সাবডিরেক্টরিতে যুক্ত হয়ে যায়।

আজকের আলোচনার ১ নম্বর সেকশনের দ্বিতীয় ছবিটা দেখুন। যখন আমার সিস্টেমে '/dev/hda6' পার্টিশনটা মাউন্ট করা রয়েছে '/mnt/slackware' ডিরেক্টরিতে, তখন আমি '/mnt/slackware' ডিরেক্টরিতে দাঁড়িয়ে '`ls`' কমান্ড দিলে আমি পাব '/mnt/slackware/bin', '/mnt/slackware/boot', '/mnt/slackware/dev', '/mnt/slackware/etc' ইত্যাদি ডিরেক্টরিগুলো। এখন ওই ছবিতে একদম নিচের ডানদিকের নলপথের স্টপককটা খোলা আছে তাই মাউন্টপয়েন্ট '/mnt/slackware' বেয়ে তথ্য চলাচল করছে। এই ডিরেক্টরিগুলো স্ল্যাকওয়ার পার্টিশনের রুট ডিরেক্টরির মধ্যেকার সাব ডিরেক্টরি, স্ল্যাকওয়ার অপারেটিং সিস্টেমে বুট করলে তাদের স্ল্যাকওয়ার সিস্টেমের রুট পার্টিশনের সাবডিরেক্টরি হিশেবেই দেখাত। তখন, স্ল্যাকওয়ার অপারেটিং সিস্টেমে, '/home/dd' ডিরেক্টরিতে কোনো ফাইল আমি লিখে থাকলে এখন সেটাকে পাব '/mnt/slackware/home/dd' ডিরেক্টরির মধ্যে একই চেহারায়।

কিন্তু মাউন্ট না-করা অবস্থায়, যখন ওই স্টপককটা বন্ধ আছে, পার্টিশনটা বিচ্ছিন্ন হয়ে গেছে মূল ঐক্যবদ্ধ ফাইলব্যবস্থা থেকে, তখন আমি ওই একই '/mnt/slackware' ডিরেক্টরিতে দাঁড়িয়ে একই '`ls`' কমান্ড দিলে দেখাবে কিছুই নেই। কারণ তখন ওই পার্টিশনের নিজস্ব ফাইলব্যবস্থা মূল ঐক্যবদ্ধ ডিরেক্টরিন্যস্ত ফাইলব্যবস্থায় যুক্ত নেই, মানে মাউন্ট করা নেই। এই মাউন্ট করা হবে কি হবে-না, কোথায় হবে, কখন হবে সেগুলো আপনার ইচ্ছে অনুযায়ী ঘটবে। আপনি যদি নিজের কোনো ইচ্ছে আলাদা করে না জানিয়ে থাকেন, তাহলে সেটা হবে সিস্টেমের ডিফল্ট বা চালু পথে।

ঠিক তেমনি যখন '/mnt/windows/c' আর '/mnt/windows/d' নলপথের স্টপকক দুটো খোলা আছে, মানে '/dev/hda1' আর '/dev/hda2' পার্টিশনদুটো মাউন্ট করা আছে, তখন '/mnt/windows/c' আর '/mnt/windows/d' ডিরেক্টরিদুটোয় পাব উইনডোজের সি-ড্রাইভ আর ডি-ড্রাইভের গোটা ফাইলব্যবস্থা। স্টপকক বন্ধ হলেই, আনমাউন্ট করা হলেই তারা হারিয়ে যাবে। তখন আমার '`ls`' কমান্ড '/mnt/windows/c' আর '/mnt/windows/d' অব্দি এসে থেমে যাবে। একই ঘটনা ঘটবে '/dev/hdb5' পার্টিশন মাউন্ট করা থাকলে '/mnt/arkive' ডিরেক্টরিতে। মানে মূল ঐকবদ্ধ ফাইলব্যবস্থা কোনো একটা পার্টিশনকে যে কোনো সময় 'তুমি যে আমার' করে নিতে পারে, মাউন্ট করে, জাহাজে ভাসমান নাবিকদের মত যে বন্দরে সেটা করছে, সেই বন্দরটা হল মাউন্ট-পয়েন্ট।

একটা পার্টিশনকে আপনি যে কোনো ডিরেক্টরিতে মাউন্ট করতে পারেন, 'mount' কমান্ড দিয়ে। আমার সিস্টেমে অটোমাউন্ট অফ করা আছে, আগেই বলেছি, এমনি এমনি সিস্টেম মাউন্ট হয়না। ধরুন, আমি আমার '/arkive/linux.books' ডিরেক্টরিতে কোনো একটা ফাইল পড়তে চাইছি, এখন আমি পাব কী করে? মাউন্ট তো করা নেই। তার মানে এখন '`ls`' মারলে '/arkive' ডিরেক্টরি তো শূন্য দেখাবে। এই অবস্থায় আমায় কমান্ড দিতে হয়, '`mount /dev/hdb5 /arkive`'। এই কমান্ডটা রুট হয়ে দিতে হয়, যদিনা অন্য কোনো বিশেষ ব্যবস্থা করা থাকে, '/etc/fstab' বলে একটা ফাইলে। আমারটায় সেটা করা আছে, সেই কথায় আসছি একটু বাদেই। এবার এই গোটা কমান্ডটার কাঠামোটা খেয়াল করুন — প্রথমে মাউন্ট কমান্ডটা, তারপর পার্টিশনের নাম, তারপর সেই ডিরেক্টরির নাম যেখানে মাউন্ট করা হবে। এটা মাথায় রাখবেন, মাউন্ট করা হয় একটা পার্টিশনকে, করা হয় একটা

ডিরেক্টরিতে। এখন থেকে ওই পার্টিশনের ফাইলদের ওই ডিরেক্টরিতে পাওয়া যাবে। আদতে গোটা কমান্ডের কাঠামোটা আর একটু বড়, 'mount -t <fstype> -o <options> <device> <dir>'। এর মধ্যে আমরা প্রথম দুটো অংশ বাদ দিয়েছি, '-t'-এর পরে '<fstype>' অংশে দিতে হয় ফাইলসিস্টেমের মানে তথ্যকাঠামোর নাম। যেমন আমরা জানি আমাদের '/dev/hdb5' পার্টিশনটায় ফাইলসিস্টেম হল রাইজারএফএস (reiserfs), এখানে সেই নামটা দিতে হত। সচরাচর এটা না-দিলেও চলে, সিস্টেম নিজেই বুঝে নেয়। '-o'-এর পরে '<options>' অংশে বিভিন্ন রকম অপশান দেওয়া যায়। যেমন, আমরা হয়ত চাইছি শুধু রিড-অনলি রকমে পার্টিশনটাকে মাউন্ট করতে, কোনো ফাইল লিখতে আমরা চাইনা, শুধু পড়তে চাই, তখন অপশান দিতে হত 'ro' মানে রিড-অনলি। কিছু বলে না-দিলে সিস্টেম মাউন্ট করে তার ডিফল্ট রকমে, মানে, তাতে ফাইল লেখা আর পড়া দুই-ই করা যায়। তার মানে, তখন কমান্ড দিতে হত 'mount -o ro /dev/hdb5 /arkive'। এখানে '<device>' তো আমরা জানিই, '/dev/hdb5' নামের পার্টিশন, আর '<dir>' মানে ডিরেক্টরি মানে মাউন্টপয়েন্ট হল '/arkive'। ম্যানপেজে দেখুন। এইভাবে আমরা মাউন্ট করে নিতে পারি যে কোনো পার্টিশনে, এমনকি একাধিক জায়গাতেও মাউন্ট করা যায় একই ডিভাইসকে। বা মাউন্ট করা যায় একটা ইমেজ ফাইলকেও। সেসব আপনি ধীরে ধীরে শিখে যাবেন, হাউটুতে দেখবেন সব দেখা আছে। এক এক করে এইরকম সবগুলো পার্টিশনেই মাউন্ট করা যায়। নিজেই করে দেখুন।

একটু এবার '/etc/fstab' ফাইলটার সঙ্গে পরিচিত হয়ে নিন। আমার মেশিনের সুজে সিস্টেমের '/etc/fstab' ফাইলটা তুলে দিলাম এখানে। খেয়াল করুন, আমার মেশিনে আরো একটা 'fstab' ফাইল আছে। কোন ফাইলটা মনে করতে পারছেন? '/mnt/slackware/etc/fstab'। কেন বলুন তো? সিস্টেমের কনফিগারেশন ফাইল একটা সিস্টেমে তো একটার বেশি থাকার কথা না, তাহলে? আসছি সেই কথায় আগে '/etc/fstab' ফাইলটা চিনুন।

```
#device     mountpoint         fs         options               dump  fsck
/dev/hdb3   /                  xfs        defaults              1     1
/dev/hdb1   /boot              xfs        defaults              1     2
/dev/hdb2   swap               swap       pri=42                0     0
/dev/hda1   /mnt/windows/c     vfat       rw,noauto,noexec      0     3
/dev/hda5   /mnt/windows/d     vfat       rw,noauto,noexec      0     3
/dev/hda6   /mnt/slackware     reiserfs   rw,noauto,user,exec   1     3
/dev/hdb5   /arkive            reiserfs   rw,noauto,exec        1     3
/dev/cdrom  /media/cdrom       auto       ro,noauto,user,exec   0     0
/dev/cdwri  /media/cdwri       auto       ro,noauto,user,exec   0     0
/dev/fd0    /media/floppy      auto       noauto,user,sync      0     0
devpts      /dev/pts           devpts     mode=0620,gid=5       0     0
proc        /proc              proc       defaults              0     0
usbdevfs    /proc/bus/usb      usbdevfs   noauto                0     0
```

এই ফাইলটা কিন্তু হুবহু আমার সুজে সিস্টেমের '/etc/fstab' ফাইলটা নয়। একটু একটু বদলাতে হয়েছে, যেমন একদম শুরুর যে লাইনটা, '#' দিয়ে শুরু, ওটা সুজেতে থাকেনা, অন্য কোনো সিস্টেমে দেখেছি, বোঝার সুবিধের জন্যে এখানে যোগ করে দিয়েছি। '#' চিহ্নটা আদতে পাঠক বদলানোর জন্যে। এইটা কোনো লাইনের গোড়াতেই থাকা মানে, কারনেলকে বলে দেওয়া, ওহে সরলমতি কারনেল, এই লাইনটা ফর অ্যাডাল্টস, তোমাকে এটা পড়তে হবেনা। এটা শুধু জ্যান্ত ব্যবহারকারীরা পড়বে, যাদের হাত পা ঠান্ডা-লাগা সংস্কৃতি ইনকাম-ট্যাক্স ইত্যাদি আছে। ফর ইয়োর আইজ ওনলি, ভো রিডার, আপনি যাতে বুঝতে পারেন, কোথায় কী আছে। একবার পড়ে দেখুন, শেষদুটো এন্ট্রি বাদ দিয়ে আর সবগুলোই আপনি বুঝতে পারছেন। তাও, একদম শেষেরটা, 'fsck', যেন চেনাচেনা লাগছে, কোথায় একটা দেখেছিলেন? আমি মনে করিয়ে দিচ্ছি, সাত নম্বর দিনের ১১ নম্বর সেকশনে — এত খারাপ মেমরি নিয়ে চলেন কী করে? হাতে একটু টাকা এলে সিস্টেম আপডেট করে নিন। চাঁদনির বাজারে স্মৃতি চিপ প্রতি কত একটা দরে পাওয়া যাচ্ছে। আর একদম শেষ তিন লাইন আপনি আপাতত গয়ায় পিণ্ডদান করে দিন, ইনস্টল করার সময়ে সিস্টেমই ওগুলো লিখেছে, এই পাঠমালাতে আসবে না ওদের আলোচনা, আর আপনার আপাতত কাজেও লাগবে না, যখন লাগবে নিজেই শিখে নিতে পারবেন। এইমাত্র যে 'mount' কমান্ডের আলোচনা করলাম আমরা তার '<options>' অংশগুলোকে এবার 'man mount' করে মিলিয়ে নিন। দেখুন, প্রত্যেকটাই আপনি হুবহু বুঝতে

পারছেন। শুধু 'mount' ম্যানপেজ পড়ে হবেনা কিন্তু 'man fstab' করে সেই ম্যানপেজটাও পড়ে নিতে হবে। একটু আগে 'mount /dev/hdb5 /arkive' কমান্ড দিয়ে আমরা যে মাউন্ট করছিলাম '/dev/hdb5' নাবিককে, '/arkive' বন্দরে, সেই কমান্ডটা আমি ছোট করে 'mount /dev/hdb5' বা 'mount /arkive' দিতে পারতাম, তখনো একই কাজ হত, কারণ মাউন্ট অন্য প্রয়োজনীয় তথ্যগুলোকে 'fstab' ফাইল থেকে নিজেই পড়ে নিত।

'/etc/fstab' ফাইলের প্রথম মানে 'device' স্তম্ভটা তো সহজেই বুঝতে পারছেন। পার্টিশন ডিভাইসগুলোর নাম। দ্বিতীয় 'mountpoint' হল সেই ডিরেক্টরি যেখানে মাউন্ট করা হবে। আজকের আলোচনার দুই নম্বর মানে বাড়ির প্ল্যানের মত দেখতে ছবিটার সঙ্গে মিলিয়ে নিন। এর প্রথম এন্ট্রিটা রুট ফাইলসিস্টেম মানে '/' হওয়াটাই প্রথা। তার পরে 'fs' মানে ফাইলসিস্টেম, তথ্য তথা ফাইল রাখার কাঠামো, ইএক্সটিটু থ্রি জেএফএস এক্সএফএস ফ্যাটথার্টিটু ইত্যাদি যাদের পিন্ডি আমরা তিন নম্বর সেকশনে চটকালাম। এই স্তম্ভে দেখুন উইনডোজ পার্টিশন দুটোর ফাইলসিস্টেমের নাম 'vfat'। এটা হল উইনডোজ ফ্যাটথার্টিটুর গ্নু-লিনাক্স বিকল্প। উইনজোজের ওই পার্টিশনগুলোকে গ্নু-লিনাক্স ওই সিস্টেমেই পড়ে। ফাইলের নিচের দিকে নয়, দশ এবং এগারো নম্বর লাইনে দেখুন, তিনটে ডিভাইস, '/dev/cdrom', '/dev/cdwri' আর '/dev/fd0'। প্রথমটা সিডি পড়ার ড্রাইভ, দ্বিতীয়টা সিডি পোড়ানোর ড্রাইভ, আর তিন নম্বরটা ফ্লপি। এই তিনটেই কিন্তু লিংক ফাইল। নিজে 'ls -al' করে দেখুন, আপনি তো জানেন, কী করে দেখতে হয়।

আপনার সিস্টেমে যদি সিডি পোড়ানোর ড্রাইভ না-থাকে, তাহলে '/dev/cdwri'-টা থাকবে না। বা এগুলো অন্য কোনো নামেও থাকতে পারে, যেমন সুজে নিজে নিজে একটা বিচ্ছিরি লম্বা নাম দেয়, '/dev/cdrecorder', লাইন ভেঙে যায়, ফাইলটা দেখতে বিচ্ছিরি লাগে। ওই নামটা বদলে নেওয়া নিজের। এবং এই '/dev/cdrecorder' নামটাও সুজে বা অন্য কোনো অপারেটিং সিস্টেমই নিজে নিজে দেয়না। পাঁচ নম্বর দিনের ৬ নম্বর সেকশনে আমরা '/etc/lilo.conf' ফাইলটা নিয়ে আলোচনা করেছিলাম। তাতে কিছু লাইন আমরা বাদ দিয়ে গেছিলাম, এরকম একটা লাইন দেখুন, সুজে আর স্ল্যাকওয়ার দুটো অংশেই একবার করে মোট দুবার আছে। 'append = "hdc=ide-scsi"'। সুজে লাইনটায় আরো একটা অংশ আছে 'splash=0'। ওটা সুজের নিজস্ব কায়দা, ভুলে যান, সুজে ব্যবহার করলেই লাগবে কেবল। বুট করার সময় থকথকে রোমান্টিক নীল রঙে পঞ্চশর ভস্ম গোটা স্ক্রিনে ছড়িয়ে দেওয়ার অভ্যেস আছে সুজের, ছবিতে ছবিতে ছয়লাপ করে দেওয়ার, সেইসব কায়দা বাজি বন্ধ করতে বলা আছে। আর 'append = "hdc=ide-scsi"' অংশটা দিয়ে সুজে আর স্ল্যাকওয়ার দুটো সিস্টেমকেই বলা আছে, '/dev/hdc' নামের যে ডিভাইসটা, যদিও সেটা আইডিই ডিভাইস, সেটাকে স্কাসি ডিভাইস হিশেবে পড়ো। এরই অন্য নাম স্কাসি-নকল বা স্কাসি-এমুলেশন। এটা করতে হয়, সিস্টেমে সিডি পোড়াতে চাইলে, আপনার সিডি-ড্রাইভটা যদি বার্নার-ড্রাইভ হয়। তখনই ওই '/dev/cdrecorder' লিংকটা সিস্টেমের ডিভাইস ফাইলে বানানো হয়, যা আপনার '/dev/hdc' ডিভাইসটাকে আঙুল দেখায়। এখন থেকে যখনই আপনি লিংকটায় কিছু করছেন, লিখছেন বা পড়ছেন, আসলে করছেন সেটা আপনার বার্নার ডিভাইসটাকে।

'/etc/fstab' ফাইলের '/dev/cdrom', '/dev/cdwri' আর '/dev/fd0' এই তিনটে ডিভাইসেই দেখুন ফাইলসিস্টেম করা আছে 'auto'। মানে আপনি সিস্টেমের বিচারবুদ্ধির উপর ছেড়ে দিচ্ছেন। রিমুভেবল মানে বার-করে-ফেলা-যায়-এমন ডিভাইসে, সিডি ফ্লপি ইত্যাদি, এটাই চালু প্রথা। ম্যানপেজ পড়ে দেখুন আপনি এখানে একটু আগে পড়া জলিয়েট রকরিজ ইত্যাদিও ব্যবহার করতে পারেন। এখন তো আপনার সিডিকাঠামো বিষয়ে গুচ্ছ ফান্ডা। এর পরে আসে 'options' স্তম্ভটা, এতে আমাদের একটাই কথা বলার আছে, কোষ্ঠকাঠিন্যের সমস্যায় খেজুর অত্যন্ত উপকারী, যদিও, অপর্যাপ্ত ব্যবহারের আগে একবার পোস্ট-পড্রিয়াল চেক করে নেওয়া ভালো। 'dump' স্তম্ভটায় আসতে পারে এক বা শূন্য। ডাম্প হল একটা ইউটিলিটি সফটওয়ার বা ব্যবহারিক প্রোগ্রামের নাম। ব্যাক-আপ করার। এই স্তম্ভে এক থাকা মানে ডাম্প সেই পার্টিশনকে ডাম্প করবে, শূন্য থাকা মানে করবে না, যখন আপনি ডাম্প চালাবেন। ডাম্প প্রোগ্রামটা আপনি চালালে, তখন কোন পার্টিশনকে কী করতে হবে সেটা ডাম্প এই 'fstab' ফাইল থেকে পড়ে নেবে। শেষ স্তম্ভ হল 'fsck'। ফাইলসিস্টেমকে চেক করা হবে কিনা, তার হাল-হকিকত পরখ করে নেওয়া হবে কিনা, হলে, কার পরে কাকে চেক করা হবে সেইটা বলে দিচ্ছে এই স্তম্ভের মানগুলো। গ্নু-লিনাক্স বুট করার সময়, যে

ফাইলসিস্টেমগুলোয় এই স্তম্ভের মান শূন্য করা নেই, তাদের চেক করে, কোনো দুর্নীতি ঘটেছে কিনা, কোথাও কোনো তথ্যকাঠামো ঘেঁটে গেছে কিনা, কোরাপশন আছে কিনা ফাইলব্যবস্থায়। আর এই স্তম্ভের সংখ্যা মেনে পরপর সেই চেকটা করে। যেমন, সচরাচর রুট পার্টিশনের থাকে এক, তার মানে প্রথম চেক করবে এই রুট পার্টিশনকে। তারপর দুই, তারপর তিন, এইভাবে। আপনার মেশিনে যখন গ্নু-লিনাক্স সিস্টেমটা ইনস্টল হয়েছিল, তখনই প্রাথমিক ‘fstab’ ফাইলটাকে বানিয়ে ‘/etc’ ডিরেক্টরিতে রেখেছিল সিস্টেম। প্রতিবার তার প্রয়োজন অনুযায়ী সেটাকে পড়ে। আপনি যদি কোনো বদল ঘটান, আপনার ইচ্ছা বা প্রয়োজন মোতাবেক, সেই বদল অনুযায়ী তাকে পড়বে। আপনি যদি কখনো ছোটখাটো পার্টিশন কাঠামো বদলান আপনার হার্ডডিস্কের, সেটাকেও আপনাকে নিজেই লিখে দিতে হবে। বা, যদি চান, কোনো একটা পার্টিশন মাউন্ট হবে কি হবে না, সেটাকেও আপনার বলে দিতে হবে। যেমন দেখুন, চার থেকে সাত নম্বর লাইনে, ‘options’ অংশে, ‘rw,noauto,user,exec’ এই অংশটা, বা এর নানা রকমফের, আমার যোগ করে দেওয়া। ‘rw’ মানে ফাইল পড়া আর লেখা দুটোই যাতে করা যায়। ‘noauto’ মানে আপনা থেকে সিস্টেম মাউন্ট করবে না, আলাদা করে কমান্ড দিয়ে করে নিতে হবে। ‘user’ মানে শুধু রুট নয়, অন্য কোনো সাধারণ ব্যবহারকারীও মাউন্ট করতে পারবে। ‘/dev/hdb5’ পার্টিশনের বেলায় দেখুন এই ‘user’ অংশটা নেই, যাতে কখনো ভুলবশত এই পার্টিশনের কোনো ফাইল কেউ বদলে না-ফেলে। ‘exec’ অংশের মানে কোনো বাইনারি ফাইল চালানো যাবে, এবং ‘noexec’ মানে তার ঠিক উল্টো। মাউন্টের প্রতিটি অপশানেই তাই, একটা অপশানের ঠিক উল্টোটা হল সেই অপশানের আগে ‘no’ যোগ করা।

এবার দেখুন তো আমার মেশিনের ‘/mnt/slackware/etc’ ডিরেক্টরি থেকে অন্য ‘fstab’ ফাইলটা এবার পড়ে দেখুন তো, কিছু বুঝতে পারেন কিনা, তফাতগুলো কোথায় ঘটছে, এবং কেন।

```
#device      mountpoint        fs          options              dump   fsck
/dev/hdb2    swap              swap        defaults             0      0
/dev/hda6    /                 reiserfs    defaults             1      1
/dev/hda1    /mnt/windows/c    vfat        noauto,user,rw       1      0
/dev/hda5    /mnt/windows/d    vfat        noauto,user,rw       1      0
/dev/hdb5    /mnt/arkive       reiserfs    noauto,user,ro       0      0
/dev/cdrom   /mnt/cdrom        iso9660     ro,noauto            0      0
/dev/cdwri   /mnt/cdwri        iso9660     ro,noauto            0      0
/dev/fd0     /mnt/floppy       auto        noauto,owner         0      0
devpts       /dev/pts          devpts      gid=5,mode=620       0      0
proc         /proc             proc        defaults             0      0
```

তাকান, এখানেও একটা ‘/’ পার্টিশন আছে, কিন্তু তার ডিভাইসটা আগের ফাইল থেকে বদলে গেছে, এখন রুট পার্টিশন হয়েছে ‘/dev/hda6’। কারণ, স্ল্যাকওয়ার সিস্টেম যখন বুট করে, তার রুট পার্টিশন হয় ওটাই। তখন সেই রুট পার্টিশনে যেটা ‘/etc’ ডিরেক্টরি, সেই ডিরেক্টরির ‘/etc/fstab’ ফাইলটাকে তার স্বাভাবিক জায়গাতেই পাই। কিন্তু সুজে দিয়ে বুট করার পর পাই ‘/mnt/slackware/etc’ ডিরেক্টরির ভিতর ‘/mnt/slackware/etc/fstab’ ফাইল হিশেবে। দুটো ‘fstab’ ফাইল মিলিয়ে দেখুন, সোয়াপ পার্টিশন একই আছে। সুজে আর স্ল্যাকওয়ার দুটো অপারেটিং সিস্টেমই এই সোয়াপ পার্টিশনটা ব্যবহার করে, যখন সুজে দিয়ে বুট করি তখন সুজে, যখন স্ল্যাকওয়ার দিয়ে বুট করি তখন স্ল্যাকওয়ার। এবং সুজের যেটা রুট পার্টিশন, ‘/dev/hdb3’, সেটা এই ‘fstab’ ফাইলে উল্লিখিতই নেই, তার কারণটা তো আগেই বলেছি, স্ল্যাকওয়ারে যে কারনেলটা আমি এখন ব্যবহার করি তার মধ্যে এক্সএফএস ড্রাইভারটা নেই, তাই সে মাউন্ট করতে পারেনা এই পার্টিশনে। ঠিক সেই একই কারণে ‘/dev/hdb1’ পার্টিশনটাও নেই। দুটো লাইন তাই কমে গেছে স্ল্যাকওয়ারের এই ‘fstab’ ফাইলে। অন্য তফাতগুলো নিজেই বুঝে নিতে পারবেন, শুধু একটা জিনিষ, স্ল্যাকওয়ার অনেক বিশুদ্ধতাপন্থী, তাই ডিফল্ট রকমে সে সিডিরমে শুধু ‘iso9660’ ফাইলব্যবস্থার মত করেই পড়ে, দরকার পড়লে রকরিজ বা জলিয়েট রকমে পড়ানো যায়। কী ভাবে? সেটা আমার চেয়ে এখন আপনারই ভালো জানার কথা। আমি সেই কবে পড়েছি, আপনি এই এইমাত্র পড়লেন। কিন্তু পড়বেন কোথা থেকে? এখানে তো দেখাই যাচ্ছে, একটা ডিস্ট্রো থেকে আর একটা ডিস্ট্রোর ডিফল্ট মানে স্বাভাবিক রকমটা আলাদা। এই টুকিটাকিগুলো আপনাকে পড়তে হবে সেই ডিস্ট্রিবিউশনের নিজের ডকুমেন্টেশন থেকে। খেয়াল রাখবেন, একটা অপারেটিং সিস্টেম থেকে আর একটা অপারেটিং সিস্টেমে এই ছোটখাট তফাতগুলো থাকেই।

৫।। একটু ছোট্ট কুপথ — ফাইল বদলানো

একটু কুপথে ঘুরে আসি, ডাইগ্রেস করে আসি, চলুন, তারপর ফের আবার আমাদের ফাইলসিস্টেমের আলোচনার লাস্ট ল্যাপে আসা যাবে। আগের সেকশনে বারবার বলছিলাম ফাইল বদলানো, সেখানে একটা জিনিষ বদলে অন্য জিনিষ লিখে দিতে হবে, ইত্যাদি। কিন্তু সেটা করবেন কী করে? আমাদের আলোচনার শূন্য নম্বর দিনে আমরা প্রচুর কথা বলেছিলাম বিশুদ্ধ টেক্সট এডিটর আর ওয়ার্ড প্রসেসর নিয়ে। তারপর আবার সেই আলোচনা এসেছিল এক আর দুই নম্বর দিনে কমান্ড এডিটরের প্রসঙ্গে। এইবার আপনার সেই কমান্ড এডিটরদের একটাকে লাগে। বিল জয়ের এক্স (ex) তথা ভিআই (vi) এডিটর আর আরএমএসের ইম্যাক্সের (emacs) কথা মনে আছে? ভিআই এখন আরো অনেক উন্নত। তার নাম ভিম (vim), ভিআই-ইমপ্রুভড, উন্নতি এবং বলশালীতা বোঝাতে দীর্ঘ-ঈকার লাগানোই যায়। তাতে দ্বিতীয় পাণ্ডবের মত আগ্রাসী হয়ে ওঠেনা, র‍্যাম সিপিইউ ইত্যাদি অত্যন্ত কম খায় এই ভীম। এ ব্যাপারে আপনি লাগ-এর অরিজিতকে মেল করতে পারেন। ভীমগীতা গেয়ে ও বেশ নাম কিনেছে। আমি নিজে এই কাজে ইম্যাক্স ব্যবহার করি। ধরুন ‘fstab’ ফাইলটা বদলাতে হবে। ‘`emacs /etc/fstab`’ কমান্ড দিয়ে ফাইলটাকে খুলি, যা দরকার এডিট করি। তারপর ‘<Ctrl><S>’ মেরে সেভ করি, এর মানে কন্ট্রোল সুইচটা টিপে রেখে ‘S’ সুইচটা টেপা, এবং কাজ শেষ হলে ‘<Ctrl><X>’ মেরে বেরিয়ে আসি। এটা কিন্তু রুট হয়ে করতে হয়। সি শিখতে গিয়ে যে মকশো করার প্রোগ্রাম বানাতে হয়, সেগুলো লিখি এবং কম্পাইল করি ইম্যাক্সে। আমার কাছে এর খুব কোনো উদ্দেশ্য নেই, ডিপ্রেশন কাটানো ছাড়া। রান্না করা আর প্রোগ্রাম লেখা এ দুটোই দেখেছি খুব কাজ দেয়। তবে রান্না করতে গেলেই বাজার করা লাগে, যেটা বোর কাজ এবং খরচেরও। আর প্রোগ্রাম লিখতে কম্পাইল করতে চালাতে আলাদা কোনো খরচই নেই, টেবিলে বসেই করা যায়। ইম্যাক্সের কাছে এই কাজটুকু হল একটা তিমি মাছের পাখনা নাড়ানোর মত। কিন্তু ইম্যাক্স ঠিক ভাবে শেখাটা নিজেই বিরাট একটা কাজ। ইম্যাক্সে ঢুকে আপনি যদি ‘<Ctrl><H>’ মারেন তাহলে আপনি ইম্যাক্সের হেল্পে ঢুকতে পারবেন, ঢুকে পড়তে শুরু করুন, যখন যতটুকু দরকার। একসঙ্গে গোটাটা পড়া একমাত্র কোনো নিরক্ষরের পক্ষেই সম্ভব, শুধু পেজ ডাউন করে গেলেই হয়। এই ভিম আর ইম্যাক্স ছাড়াও জো (joe) আছে, সাতকেলে এড (ed) আছে। এদের শেখা শুরু করার জন্যে ম্যানপেজ আছে।

এবার ধরুন, আপনি আপনার সিস্টেমে দেখলেন কোনো কোনোটা নেই, এদিকে আপনি সেটা দিয়ে কাজ করতে চান। তখন আপনার ডিস্ট্রিবিউশন সিডি, যা দিয়ে আপনি ইনস্টল করেছিলেন, সেটায় খুঁজে দেখুন। এরপরেও হয়ত দেখলেন নেই, বা যে কাজটা করার জন্যে যে প্যাকেজের যে ভার্শনটা চাইছেন সেটা নেই। ধরুন যেমন ‘rman’, একটু আগে যেটা ব্যবহার করার কথা বললাম, সেটা সব ডিস্ট্রোয় থাকেনা। বা সমগ্র হাউটু-টা (howto)। বা অর্বুদ অর্বুদ লিনাক্সীর প্রাণের চেয়ে প্রিয় এমপ্লেয়ার। সেগুলো কী করে পাবেন? নেটে যদি পেতে চান, তার একটা ভালো জায়গা ‘www.freshmeat.net’। বা ‘www.sourceforge.net’ । এই সাইটে গিয়ে আপনার সাধের প্যাকেজটার নাম দিয়ে সার্চ দিন। তারপর হাইপার লিংক দিয়ে সেটাকে ডাউনলোড করুন। এই ইনস্টলার ফাইলগুলো সচরাচর ‘*.rpm’ বা ‘*.tar.bz2’ বা ‘*.tar.gz’ বা ‘*.tgz’ হয়।

এই আরপিএম ব্যাপারটা চালু করেছিল রেডহ্যাট, আরপিএম এসেছে রেডহ্যাট প্যাকেজ ম্যানেজমেন্ট থেকে। পরে এই ধরনের ফরম্যাট ম্যানড্রেক বা সুজে ব্যবহার করে। আরো কেউ কেউ করে। এই ‘*.rpm’ ফাইলগুলো ইনস্টল করার জন্যে মূল একটা কমান্ড হল ‘rpm’। ‘rpm’ কমান্ডের ম্যানুয়াল পড়ে নিন। যেমন আপনি রুট হয়ে কমান্ড দিতে পারেন ‘`rpm -Uvh mundu.rpm`’। এটা আপনার সিস্টেমে ‘mundu.rpm’ প্যাকেজটাকে ইনস্টল করে দেবে, মানে ‘mundu’ নামের প্রোগ্রামটাকে। বা, আগেই ইনস্টল করা থাকলে আপডেট করে দেবে। যারা একটা কার্যোপযোগী মুণ্ডের অভাবে ভোগেন, এই প্যাকেজটা তাদের খুবই কাজ দেয় বলে শুনেছি। ‘*.tgz’ হল স্ল্যাকওয়ারের নিজের প্যাকেজ বানানোর স্টাইল। অন্য অনেক ডিস্ট্রোতেও ব্যবহার করা যায়। একধরনের প্যাকেজ থেকে অন্য ধরনের প্যাকেজে বদলে নেওয়ার একটা ভালো প্রোগ্রাম আছে, তার নাম ‘alien’। ‘*.deb’ হল ডেবিয়ান ডিস্ট্রোর প্যাকেজ বানানোর রকম। যে ডিস্ট্রো আপনি ব্যবহার করছেন তার থেকে ভালো করে পড়ে নিন।

এই যে ধরুন একটা প্রোগ্রাম আপনি ইনস্টল করছেন, এর মানে কী? এই প্রোগ্রামটা লেখা হয়েছে কিছু কোড দিয়ে। সেই কোডটা কম্পাইলড হয়ে বাইনারি ফাইল তৈরি হয়। এই বাইনারি ফাইলগুলো হল এক্সিকিউটেবল, এদের চালানো যায়। এবার কোডকে কম্পাইল করে বাইনারিটা তৈরি হয়ে গেছে, একে বলে প্রিকম্পাইলড বাইনারি। এই

আরপিএম ব্যাপারটা হল তাই। আর মূল কোডটাকে নিজের মেশিনে চালিয়ে কম্পাইল করে নেওয়াও যায়। কম্পাইল করার জন্যে দরকার যে কম্পাইলারটা, তাদের একটা গোটা সংগ্রহ তো যে কোনো গ্নু-লিনাক্স ডিস্ট্রিবিউশনের এমনিতেই দেওয়া থাকে — জিসিসি (gcc)। এবার মূল কোডটাকে কম্পাইল করে নিলেই হল। এই কোডগুলো দেওয়া থাকে '*.tar.bz2' বা '*.tar.gz' ফাইলে। '*.tar.*' মানে সিন্দুক ফাইল, টার প্যাকেজ দিয়ে বানানো, আগেই বলেছি। আর 'bz2' মানে তাকে কোঁকড়ানো হয়েছে 'bzip' প্রোগ্রাম দিয়ে, আর 'gz' মানে কোঁকড়ানো হয়েছে 'gzip' দিয়ে। ম্যানপেজ পড়ে নিন। নানা ভাবে এদের নিয়ে কাজ করা যায়, ইউনিক্স তথা গ্নু-লিনাক্সের দর্শনই তাই, কোনো একটা কাজ সবসময়েই একাধিক ভাবে করা যায়। 'tar' দিয়ে সরাসরি এই 'bz2' বা 'gz' ফরম্যাটে বানানো কোঁকড়ানো সিন্দুক খোলার উপায় আছে। ধরুন ওই 'mundu.rpm' এবার আপনি আর 'rpm' মানে প্রিকম্পাইলড বাইনারিতে না-নামিয়ে কোঁকড়ানো সিন্দুকে নামিয়েছেন। আগেরবার ইনস্টল করার পর থেকে বহুত গন্ডগোল করছিল, আপনার নৃমুণ্ডের আবছায়া তৈরি হচ্ছিল, আলো অন্ধকারে যেখানেই যান, এবার আপনি আর মুন্ডু নিয়ে ওইসবে রিস্কে যেতে চাননা। আপনার মুন্ডু যাতে আপনার হার্ডওয়ারের সঙ্গে খাপে খাপে মিলে যায়, আপনার হার্ডওয়ারের সফটওয়ারের প্রতিটি খুঁটিনাটি নিয়ে আপনার সিস্টেমেই কম্পাইলড হয়, সেই ব্যবস্থাই এবার করতে চান, মানে সরাসরি ধরো আর কম্পাইল করো। ও নিজেই আপনার সিস্টেমের খুঁটিনাটি অনুযায়ী নিজেকে বদলে নেবে। ফাইলটা যদি 'mundu.tar.bz2' হয় তাহলে কমান্ড দিতে হবে, '`tar xvjf mundu.tar.bz2`'। এতে 'mundu.tar.bz2' ফাইলটা কোঁকড়ানো অবস্থা থেকে বেরিয়ে একটা স্বাভাবিক টার-সিন্দুকে পরিণত হল, এবং তারপর একটা 'mundu' নামের ডিরেক্টরি বানিয়ে, ওই সিন্দুক খুলে সমস্ত ফাইলকে সেই 'mundu' ডিরেক্টরিতে রাখা হয়ে গেল ওই একটা কমান্ড দিয়েই। যদি ফাইলটা 'mundu.tar.gz' হত তাহলে আপনার কমান্ডটা সামান্য একটু বদলে যেত, হত '`tar xvzf mundu.tar.gz`'। এতেও ওই একই ভাবে আপনি 'mundu' বলে একটা ডিরেক্টরি পেতেন, যার মধ্যে ফাইলে বা সাবডিরেক্টরিতে ভরা থাকত আপনার প্রয়োজনীয় কোডগুলো।

এবার ওই 'mundu' ডিরেক্টরিতে ঢুকে আপনি পাবেন একটা 'README' ফাইল বা একটা 'INSTALL' ফাইল বা দুটোই। এদের 'less' দিয়ে পড়ে নিলেই আপনি জানতে পারবেন কী করতে হবে। করাটাও এমন হাতি ঘোড়া কিছু না। এরা প্রত্যেকেই ব্যবহার করে জিসিসির মত আর একটা গ্নু পরশপাথর, যার নাম মেক (make), শীতকাল চলে যাচ্ছে দ্রুত গিয়ে খেজুরগাছে হাঁড়ি বেঁধে আসুন। এই 'make' যাতে কাজ করতে পারে তার জন্যে একটা 'makefile' দেখুন রয়েছে ডিরেক্টরিতে। এবার আপনাকে কমান্ড দিতে হবে, '`./configure`'। এই '`.`' মানে কারেন্ট ডিরেক্টরি, যে ডিরেক্টরিতে এই মুহূর্তে আপনি আছেন, সেখানে থাকা '`configure`' নামের প্রোগ্রামটা চালাও। এই কনফিগার প্রোগ্রামটা চালিয়ে ওই সফটওয়্যার প্যাকেজ ইনস্টল হওয়ার আগে আপনার সিস্টেমের প্রত্যেকটা প্রয়োজনীয় খুঁটিনাটি কনফিগারেশন জেনে নেবে। তারপর, কনফিগারেশনের কাজ শেষ হলে, কমান্ড প্রম্পট ফেরত এলে কমান্ড দেবেন '`make`'। অনেকসময় কনফিগার প্রোগ্রামটা থাকেনা, সরাসরি মেকফাইল থাকে। তখন মেক দিয়েই কাজ শুরু করতে হয়। এতে ওই রাশিরাশি কোডের ঠিক ঠিক প্রয়োজনীয় অংশ আপনার সিস্টেমের মানানসই রকমে সাজিয়ে দেবে। এবার মেকের কাজ শেষ হলে রুট হবেন আপনি আর কমান্ড দেবেন '`make install`'। গোটা কাজটা একবার কমান্ড দিয়েও করা যেত, '`./configure && make && make install`'। এই '`&&`' অংশটা ব্যাশকে বলে দিত, আগের কাজটা শেষ হলেই কেবল পরের কাজটায় যেও, ব্যাশের ম্যানপেজ থেকে মিলিয়ে নিন। এরপর বেশ কিছুটা সময় ধরে কনসোলের পর কনসোল জুড়ে রাশিরাশি প্রোগ্রামিং কচকচি ফুটে উঠতে থাকবে। তারপর কমান্ড প্রম্পট ফেরত আসবে। আর কী, খেল খতম, কম্পাইলেশন হয়ে গেল। এখুনি যান, তাদের প্রত্যেককে একটা করে মেল করে দিন, আসুন, যারা এতদিন ধরে ব্যাঙ্গবিদ্রূপ করে আসছে যে আপনার কোনো মুন্ডু নেই। মুন্ডু হাত পা যা যা খুশি এভাবে নেট থেকে নামান আর ইনস্টল করে নিন। শুধু টেলিফোন বিলটা মাথায় রাখলেই হল।

এবার আমরা যাব আমাদের সেই দীর্ঘ প্রতীক্ষার গ্নু-লিনাক্স ফাইলসিস্টেম হায়েরার্কিতে। শুধু একটা কথা বলতে ভুলে গেছি, ৪ নম্বর সেকশনে, সেটা হল, ওই 'mount' কমান্ডটার একটা বিপরীত কমান্ড আছে 'umount', যা মাউন্টের ঠিক উল্টোটা করে, মানে আনমাউন্ট। আদতে কমান্ডটার নাম ছিল 'unmount', ১৯৭০ অব্দি ইউনিক্স জগতে এই চেহারাতেই তাকে পাওয়া যেত, তার পর কবে 'n'-টা খসে গেছে, দীর্ঘ কঠোর তদন্তের পরেও জানা যায়নি 'n'-এর এন্ড ঘটল কিভাবে, কোন আততায়ীর হাতে।

৬।। গ্নু-লিনাক্স ফাইলসিস্টেম হায়েরার্কি

আগের দুদিন থেকে আজ এই দীর্ঘ আলোচনার মধ্যে দিয়ে আমাদের কাছে হার্ড ডিস্কের ভৌত এবং মূর্ত ভূমি, সেখানে তথ্য তথা ফাইল এবং ডিরেক্টরি ইত্যাদি নানা জাতের তথ্য-সমাহার রাখার প্রক্রিয়া, আর সেই তথ্যকে একটা কেজো চলন্ত কম্পিউটারের কাজের সম্পর্কে মিলিয়ে নেওয়ার জন্যে ফাইল আর ডিরেক্টরি দিয়ে সাজানো একটা বিমূর্ত ভূমি — এই তিনের আলাদা অস্তিত্ব এবং একই সঙ্গে অঙ্গাঙ্গী সম্পর্ক — এই গোটাটাই এখন আমাদের কাছে পরিচিত। মূর্ত ভূমিটা হল সেই কাঁচা ডিস্ক যা আমরা কিনে আনি। আর রুট ডিরেক্টরি থেকে হোম বিন এটসেট্রা ইত্যাদিতে বিন্যস্ত কেজো কম্পিউটার ভূমিটা হল বিমূর্ত জায়গা যা আমাদের তৈরি করে নিতে হয়। বিমূর্ত এবং সাঙ্কেতিক, বহুবার বলেছি আমরা, অ্যাবস্ট্রাক্ট এবং সিম্বলিক। এই জায়গাটা অত্যন্ত গুরুত্বপূর্ণ, কারণ যে প্রোগ্রামগুলো নিয়ে আমরা কাজ করি তারা কেউ কাজ করে একদম ফাইল নিয়ে, যেমন ধরুন একটা 'emacs' বা 'mkdir', আবার কেউ কাজ করে একদম কাঁচা ভৌত ভূমি নিয়ে যেমন 'fdisk' বা 'fsck'। আমরা যে প্রোগ্রামগুলোকে চিনি এবং আমাদের কম্পিউটার সময়ের প্রায় গোটাটাই যাদের নিয়ে কাজ চলে তারা প্রায় সবাইই কাজ করে বিমূর্ত ভূমিটায়। কিন্তু এই বিমূর্ত ভূমিটা দাঁড়িয়ে আছে তথ্য আর তথ্য-সমাহারের অনেকগুলো প্রক্রিয়ার উপর, মানে ফাইলসিস্টেম। এই ফাইলসিস্টেমগুলোর উপর দাঁড়িয়ে এরা কাজ করে, তাই এদের সঙ্গে মিলিয়ে না-নিলে গোটা প্রক্রিয়াটাই বোঝা সম্ভব না।

ইতিমধ্যেই কাজের ওই বিমূর্ত ভূমিটার সঙ্গে অনেকটাই পরিচিত, যেটা একটা ঐক্যবদ্ধ হায়েরার্কিন্যস্ত ফাইলসিস্টেম, ডিরেক্টরি সাবডিরেক্টরি ফাইলের ক্রমানুসারী সমগ্র, যা তৈরি হয়েছে বিভিন্ন তথ্য রাখার প্রক্রিয়া মানে ফাইলসিস্টেম নিয়ে। এখন থেকে আগের বাক্যে ফাইলসিস্টেম কথাটার দ্বিতীয় ব্যবহারটাকে আমরা আপাতত ভুলে যাব, যেখান থেকে শুরু করেছিলাম ছয় নম্বর দিনে, সেখানে ফেরত যাব, ফাইলসিস্টেম মানেই ডিরেক্টরি আর ফাইলের ওই ঐক্যবদ্ধ সমগ্রটা। সেটার হায়েরার্কিটাকেও চিনি আমরা। রুট ডিরেক্টরি মানে '/' থেকে শুরু। তার মধ্যে নানা ডিরেক্টরি। তাদের মধ্যে আবার ডিরেক্টরি, আবার, এই ভাবে চলে। প্রতিটি ডিরেক্টরিতেই থাকতে পারে আরো নিচের ডিরেক্টরি, মানে সাবডিরেক্টরি, যতখুশি, এবং যতখুশি ফাইল। এবার, এই ক্রমটা আছে বলেই, ঐক্যটা আছে বলেই, একটা ফাইলসিস্টেমে প্রতিটা ফাইল এবং ডিরেক্টরির একটা ঠিকানা আছে — একটা বিমূর্ত ঠিকানা, রুট ডিরেক্টরি থেকে ঠিকানাটা শুরু হয়ে শেষ হয় যে ডিরেক্টরিতে ফাইলটা আছে সেই ডিরেক্টরির নামে এসে (মূর্ত ঠিকানাটা, খেয়াল করুন, আর আমাদের আপাতত দরকার পড়বেনা, মানে কত নম্বর সিলিন্ডারের কত নম্বর সেক্টরের কত নম্বর ব্লক ইত্যাদি)। যেমন ধরুন '/mnt/slackware/etc/fstab' ফাইলটার ঠিকানা হল '/' ডিরেক্টরির মধ্যে 'mnt' ডিরেক্টরির মধ্যে 'slackware' ডিরেক্টরির 'etc' ডিরেক্টরির মধ্যে 'fstab' নামের ফাইল। আমি এবং আমার জ্যান্ত অপারেটিং সিস্টেম দুজনেই এটাকে চিনে নিতে পারব বলামাত্র। এবার পরের আলোচনায় যাওয়ার আগে একবার, মন দিয়ে, ছয় নম্বর দিনের ৪ নম্বর সেকশনের আর সাত নম্বর দিনের ১১ নম্বর সেকশনের ফাইলসিস্টেমের ছবিদুটো মিলিয়ে মিলিয়ে, তাদের পার্থক্য এবং মিল সহ, ভালো করে দেখে নিন। এবার আমরা গ্নু-লিনাক্সে ফাইলসিস্টেম হায়েরার্কির ওই ছকটা বুঝব।

ওই ছকটা কিন্তু একদিনে গড়ে ওঠেনি। '৯০-এর দশকের মাঝামাঝি অব্দি, গ্নু-লিনাক্স বয়োপ্রাপ্ত হয়ে ওঠার আগে অব্দি প্রতিটি আলাদা আলাদা ডিস্ট্রো ছকটা বানিয়ে নিত নিজের মত করে। এতে সমস্যাটা বুঝতেই পারছেন। একটা ছকে অভ্যস্ত কেউ অন্য ছকে কাজ করতে গেলে ছক বোঝাটাই তার কাজ হয়ে দাঁড়াবে। তার চেয়েও সমস্যা অবলা কোডগুলোর। তাদের বানানো হয়েছে একটা ছকের সঙ্গে মিলিয়ে — এই ডিরেক্টরিতে খোঁজো এই ফাইলটা আছে কিনা, ওই ডিরেক্টরিতে খোঁজো ওই ফাইলটা আছে কিনা, ইত্যাদি। তারা আর অন্য ছকের কোনো ডিস্ট্রোতে কাজই করবে না। প্রতিটা ডিস্ট্রোর আলাদা আলাদা প্রোগ্রাম বানাতে হবে। ইউনিক্সের জগতে পোজিক্স যে কারণে এসেছিল। সেই পোর্টেবিলিটি, স্থানান্তরযোগ্যতা, আমাদের এই আলোচনায় আমরা অনেকটা সময় ব্যয় করেছি সেটা বুঝতে। এই জন্যে এল একটা ডকুমেন্ট, তার নাম 'লিনাক্স ফাইলসিস্টেম স্ট্রাকচার' এবং সেই ডকুমেন্টটা বানানোর জন্যে একটা গ্রুপ — এই ডকুমেন্ট আর গ্রুপ দুটোকেই ডাকা হয় এফএসস্ট্যান্ড (FSSTND) বলে, ফাইলসিস্টেম স্ট্যান্ডার্ড। এই এফএসস্ট্যান্ড-এর কাজটাই বাড়তে বাড়তে পরে ড্যানিয়েল কুইনলানদের ওই ফাইলসিস্টেম হায়েরার্কি স্ট্যান্ডার্ডে পৌঁছয়, যার কথা আমরা আগেই বলেছি। এই কাজটা কিন্তু এখনো সম্পূর্ণ নয়, এখনো অনেক না-মেটা মনান্তর

আছে। যেমন যেসব স্ক্রিপ্ট (মানে শেলে কাজ করা প্রোগ্রাম) কোনো একটা বিশেষ নির্মাণের কম্পিউটারের সঙ্গে সম্পৃক্ত নয়, যে কোনো মেশিনেই চালানো যায় যাদের, তারা কি '/usr/share' ডিরেক্টরিতে থাকবে না থাকবে-না? এই '/usr/share' ডিরেক্টরিটাকে কি আরো সাবডিরেক্টরিতে ভাঙা হবে, ইত্যাদি। এরকম আরো কিছু। একটা ডিস্ট্রো থেকে আর একটা ডিস্ট্রো-তে কিছু কিছু তফাত তো প্রথম দৃষ্টিতেই পরিদৃশ্যমান। যেমন, বহিরাঙ্গিক পার্টিশনগুলোকে, যেমন আমার মেশিনে ওই চারটে পার্টিশন — '/dev/hdb5', '/dev/hda6', '/dev/hda1', '/dev/hda5', মানে আর্কাইভ, স্ল্যাকওয়ার, আর উইনডোজ পার্টিশনদুটো — এদের কোন মাউন্টপয়েন্টে মাউন্ট করা হবে এর কোনো ধরাবাঁধা সংস্থান এখনো নেই। এইরকম আরো আছে। একাধিক ডিস্ট্রোয় কাজ করতে গেলেই দেখবেন। এবং এখন আমরা যে ছকটা দেব সেটা যে প্রত্যেকটা ডিস্ট্রো হুবহু মেনে চলে তা কিন্তু নয়। এবার ছকটা দেওয়া যাক।

৬.১।। রুট ডিরেক্টরি বা '/'

এফএসস্ট্যান্ড মানার জন্যে রুট ডিরেক্টরিতে কিছু ডিরেক্টরি থাকতেই হবে, বা মূল ডিরেক্টরিটা অন্য কোথাও থাকলেও তার সিম্বলিক লিংক এখানে থাকতেই হবে, এরা হল রুট ডিরেক্টরির ন্যূনতম বাসিন্দা। গ্নু-লিনাক্সের ফাইলসিস্টেম স্ট্যান্ডার্ড অনুযায়ী এদের থাকতেই হবে। এর বাইরে আদতে আরো বহু কিছু থাকে। তাদের কথায় আসছি আমরা।

/bin — এই বিন ডিরেক্টরিতে থাকবে অত্যাবশ্যক কমান্ড বাইনারিগুলো, যাদের কমান্ড প্রম্পটে টাইপ করে তারপর এন্টার মেরে আমরা চালাই। আমরা কমান্ড প্রম্পটে কাজ করার জন্যে যে আদেশগুলো নিয়ে এই পুরো পাঠমালায় আলোচনা করেছি, তাদের প্রায় সকলেরই আবাস এই বিন ডিরেক্টরি। বিন নিয়ে আলোচনায় আসছি।

/boot — এই বুট ডিরেক্টরিতে থাকে সিস্টেম বুট করার জন্যে প্রয়োজনীয় কারনেল ইমেজ এবং তার কিছু আনুষঙ্গিক ফাইল। তারা যদি হুবহু এখানে না-থাকে, থাকে তাদের সিম্বলিক লিংক। কারনেলটা ব্যবহার করতে গেলে যে মডিউলগুলো লাগবে তাদের তালিকাও দেওয়া থাকে এখানেই। যেমন '/boot/vmlinuz'। শুধু বুট-প্রক্রিয়ার কনফিগারেশন ফাইলগুলো এখানে থাকেনা, দেখেছেন আগেই, লিলোর কনফিগারেশন ফাইল 'lilo.conf' থাকে '/etc' ডিরেক্টরিতে। তাকে যখন '`lilo`' কমান্ড দিয়ে ইনস্টল করা হয় সে কিছু ফাইল তৈরি করে এই বুট ডিরেক্টরিতে। যেমন '/boot/boot.0300' মানে মাস্টার বুট রেকর্ডের ব্যাকআপ। বা '/boot/boot.b' মানে মূল বুট সেক্টর বা তার সিম্বলিক লিংক।

/dev — এই ডেভ ডিরেক্টরিতে থাকে রাশিরাশি ডিভাইস ফাইল, যাদের নিয়ে যথেষ্ট বিশদভাবে আলোচনা হয়েছে আমাদের ছয় নম্বর দিনে।

/etc — এই ইটিসি বা এটসেট্রা ডিরেক্টরিতে থাকে একটা সিস্টেমের হার্ডওয়ার এবং সফটওয়ার সংক্রান্ত প্রায় সমস্ত কনফিগারেশন ফাইল। এর কিছু ফাইলকে এর মধ্যেই বেশ ধরে ধরে পড়েছি আমরা, বারবার। সিস্টেমের স্নায়ুকেন্দ্র বলা চলে এই '/etc' ডিরেক্টরিকে। আমরা এর কথায় পরে আসছি।

/lib — এই লিব ডিরেক্টরিতে থাকে লাইব্রেরি ফাইলগুলো, যাদের কথা আমরা আগেই বলেছি, প্রোগ্রামগুলো এই লাইব্রেরিদের কাজে লাগায়। প্রতিটি প্রোগ্রামের জন্যে আলাদা আলাদা ফাংশন বা ক্রিয়া না লিখে, সব প্রোগ্রামের জন্যে একত্রে এই লাইব্রেরি ফাংশনগুলোকে তুলে রাখা হয় লিব ডিরেক্টরিতে। আর থাকে কারনেল মডিউলগুলো, মানে, কারনেলের চলার জন্যে ছোট ছোট টুকরো প্রোগ্রাম বানানো থাকে, এক ধরনের ড্রাইভার, এক এক ধরনের উপাদানের জন্যে। ধরুন একটা বিশেষ সাউন্ডকার্ড আপনি ব্যবহার করছেন, তার জন্যে কারনেলে একটা বিশেষ মডিউল লাগিয়ে নিতে হচ্ছে। আবার সাউন্ডকার্ডটা বদলালেন, মডিউলটা বদলে নিতে হল।

/mnt — এই এমএনটি বা মাউন্ট ডিরেক্টরির সঙ্গে আপনারা ইতিমধ্যেই বারংবার পরিচিত হয়েছেন। বহিরাগত পার্টিশনদের বসতে দেওয়ার জেনেরাল শীতলপাটির মত। তবে সব ডিস্ট্রো এটা হুবহু মানেনা। ম্যানড্রেক স্ল্যাকওয়ার খুব মানে। কিন্তু সুজে নানা অন্য অন্য জায়গা বানিয়ে দেয়। যেমন আমার উইনডোজ ডিরেক্টরি দুটোর জন্যে নিজে থেকেই বানিয়ে দিয়েছিল '/data1' আর '/data2' বলে দুটো ডিরেক্টরি, একদম রুট ডিরেক্টরিতে। পরে নিজে বদলে নিতে হয়। তবে খেয়াল রাখবেন, এইসমস্ত যাবতীয় জিনিষ বদলানোর পরেই '/etc/fstab' ফাইল থেকে একবার মিলিয়ে নেবেন, সবগুলো ঠিকঠাক আছে কিনা। নিজের ইচ্ছেমতন বাড়তি মাউন্টপয়েন্ট বানিয়ে নেওয়া যায় এই '/mnt' ডিরেক্টরিতে, আমরা দেখেছি, '/mnt/arkive' বা '/mnt/slackware' ইত্যাদি। আর চালু

কিছু মাউন্টপয়েন্ট এখানে এমনিতেই থাকে, যদি ডিস্ট্রো সেখানে নিজস্ব কেরদানি না-মারে। যেমন '/mnt/cdrom' বা '/mnt/floppy' ইত্যাদি। সুজেতে সেসব থাকে '/media' ডিরেক্টরির মধ্যে।

/opt — অপশনাল বা ঐচ্ছিক, ডিস্ট্রোর ডিফল্ট ইনস্টলেশনের নিজস্ব কাঠামোর বাইরে থেকে যুক্ত প্যাকেজদের জায়গা। যেমন ওপেনঅফিস, মোজিলা ইত্যাদি প্যাকেজগুলো এখানে থাকে। এফএসস্ট্যান্ড অনুযায়ী, সমস্ত বহিরাগত প্যাকেজদেরই এখানে থাকার কথা। আর এই প্যাকেজটা ইনস্টল করার জন্যে আপনার যে আনুষঙ্গিক ফাইলগুলো থাকবে তাদের থাকার জায়গা এই '/opt' ডিরেক্টরির মধ্যেই ওই প্যাকেজের নামে আলাদা একটা ডিরেক্টরিতে। ধরুন ওই 'mundu' প্যাকেজটা, বাইরে থেকে যোগ করা প্যাকেজ, কারণ, ডিফল্টে মানুষের একটা মুন্ডু এমনিতেই থাকার কথা, আপনি যদি ইনস্টল করেন, সেই ইনস্টলেশনটা ঘটবে '/opt/mundu' ডিরেক্টরির মধ্যে। ইনস্টল করতে গিয়ে যা যা বাড়তি ফন্ট মানে অক্ষর-আকার লাগবে, বা ছবি, বা ডেটাবেস, সে সবই থাকবে এই ডিরেক্টরির ভিতর। এই প্যাকেজ যে বাইনারি ফাইল তৈরি করবে সেটা থাকার কথা '/opt/bin' ডিরেক্টরিতে, ডকুমেন্টেশন থাকার কথা '/opt/doc' ডিরেক্টরিতে, ইত্যাদি। তবে, বেশিরভাগ প্যাকেজই এতসব খাবলা দায়িত্ব পালন করেনা।

/sbin — এর মধ্যে থাকে অত্যাবশ্যক সিস্টেম বাইনারিগুলো। গ্নু-লিনাক্সে বাইনারি বা এক্সিকিউটেবল মানে চালানো যায় এমন ফাইলগুলোকে দুটো ভাগে ভাগ করা হয়। একটা সাধারণ বা নরমাল এক্সিকিউটেবল। যা সর্বসাধারণের ব্যবহারযোগ্য। আর অন্যগুলো হল ভিআইপি এক্সিকিউটেবল। সিস্টেম অ্যাডমিনিস্ট্রেশন বা পরিচালনার কাজে ব্যবহারযোগ্য বিশেষ বাইনারিগুলো থাকে এই '/sbin' ডিরেক্টরিতে। এর চেয়ে একটু কম গুরুত্বপূর্ণ সিস্টেম বাইনারিগুলো থাকে '/usr/sbin' ডিরেক্টরিতে বা '/usr/local/sbin' ডিরেক্টরিতে। এই '/sbin' ডিরেক্টরি তথা সিস্টেম বাইনারিদের কথায় আসছি আমরা।

/tmp — অস্থায়ী বা টেম্পোরারি ফাইলদের জায়গা। একটা নির্দিষ্ট সময়ের পরে যাদের আর দরকার পড়বেনা। অনেক প্রোগ্রাম কাজ করার সময় বিশেষ কিছু ফাইল বা ডিরেক্টরি বা প্রোগ্রামকে লক করে দেয়, চলার সময়টুকু জুড়ে অন্য কেউ তাদের স্পর্ষ করতে পারবেনা। কিম্বা চলার কাজে কিছু অস্থায়ী তথ্য তৈরি করে, প্রোগ্রাম চলা শেষ হয়ে যাওয়ার পরে যাদের আর কাজে লাগবে না। এই সবকিছুকে অস্থায়ী ভাবে রাখা হয় এই '/tmp' ডিরেক্টরিতে। 'su' করে রুট হয়ে নিয়ে এখান থেকে ফাইল ওড়ানোই যায়, কিন্তু কদাচ নয়, যদি-না, হুবহু জানা থাকে, ঠিক কী করছি। সেই মুহূর্তে যে প্রসেসগুলো চলছে সিস্টেমে তাদের অনেকেরই অত্যন্ত জরুরি মালপত্তর এখানে থাকে। গোটা সিস্টেম হ্যাং করে যেতে পারে ক্র্যাশ করে যেতে পারে — গ্নু-লিনাক্স সিস্টেমে যা সত্যিই একটা দুর্ঘটনা, এত কম ঘটে। যেমন ওপেনঅফিস ১.০ ভার্শনে আমার এনভিডিয়া ড্রাইভারের সঙ্গে কিছু একটা গোলযোগ হচ্ছিল, পরে, এনভিডিয়া ড্রাইভারের একটা পুরোনো ভার্শন ব্যবহার করে যেটা চলে গেল, সেই সময়ের গোটা পাঁচেক হ্যাং আজ পর্যন্ত এই দু-বছরে আমার পাঁচমাত্র হ্যাংদোলন। নিজে লিখতে গিয়ে তার আগের উইনডোজের বছরগুলোর সঙ্গে মিলিয়ে নিজেরই কেমন অবিশ্বাস হচ্ছে। এই '/tmp' ডিরেক্টরির মোট ফাইলের আয়তন সচরাচর কয়েক কিলোবাইটের বেশি হয়না। বুট বা শাটডাউনের সময় সচরাচর এই ডিরেক্টরিকে ফাঁকা করে দেওয়া হয়।

/usr — মূল '/' হায়েরার্কির মধ্যে দ্বিতীয় একটা হায়েরার্কি বা ক্রম। এই '/usr' ডিরেক্টরিতে থাকে গোটা সিস্টেমের মোট ফাইল এবং ডিরেক্টরির সমগ্রের সিংহভাগ। আমরা এখানে সুজে এবং স্ল্যাকওয়ার দুটো ডিস্ট্রোরই রুট ডিরেক্টরিতে থাকা সাবডিরেক্টরির গুলোর আলাদা আলাদা ভাবে মোট আয়তন আর গোটা সিস্টেমটার মোট আয়তন দিয়েছি, পাশাপাশি। সাধারণ একটা গড় ইনস্টলেশন। তার থেকে সামান্য বেশি হতে পারে, সবসময়ে ডিভলপমেন্টের, মানে কম্পিউটার ভাষার আর প্রোগ্রামিং-এর যে প্যাকেজগুলো এই দুটো সিস্টেমেই ইনস্টল করা আছে, তার সবগুলো সব সিস্টেমে থাকেনা। আর পাবলিশিং সংক্রান্ত প্যাকেজও দু-চারটে বেশি থাকতে পারে। এর বাইরে একটা স্বাভাবিক একটা হোম পিসির একটা সাধারণ ইনস্টলেশনের আয়তনটাও এখান থেকে পেতে পারবেন। যে কমান্ড দিয়ে আমি এ তালিকাটা পেয়েছি সেটা হল '/' ডিরেক্টরিতে দাঁড়িয়ে 'du -chs *', তারপর তাকে আমাদের চেনা কায়দায় রিডাইরেক্ট করে। ম্যানপেজ পড়ে দেখুন এই 'du' কমান্ডটা ভারি চমৎকার ভাবে ডিস্কভুমির ব্যবহারটা দেখায়। এর '-chs' অপশানগুলোর মানে নিজেই দেখে নিন। এছাড়া আর একটা

কমান্ডও এই কাজে খুব ব্যবহার হয়, 'df'। তালিকাটায় 'M' মানে মেগাবাইট, 'K' মানে কিলোবাইট, 'G' মানে গিগাবাইট। সুজেতে দেখুন '/usr' জুড়ে আছে গোটা সিস্টেমের মোট ব্যবহৃত ডিস্কভূমির প্রায় একাত্তর শতাংশ আর স্ল্যাকওয়ারে প্রায় তিরাশি শতাংশ। এই '/usr' ডিরেক্টরির আলোচনায় আমরা পরে আসছি।

/var — ভ্যারিয়েবল বা পরিবর্তনশীল তথ্যদের আবাস। সিস্টেম লগিং সংক্রান্ত তথ্যের ফাইল, মেল আর প্রিন্টার স্পুলের ফাইল, এই সব থাকে '/var' ডিরেক্টরিতে। এই '/var' ডিরেক্টরি দুটো সিস্টেম কখনো শেয়ার করতে পারেনা, মানে দুজনেই আলাদা ভাবে একই পার্টিশনকে ব্যবহার করতে পারেনা। কারণ এখানে থাকা তথ্যের গোটাটাই হল সিস্টেমের একান্ত নিজস্ব। এই ডিরেক্টরির তথ্যগুলো সিস্টেম ব্যবহারের সঙ্গে তাল মিলিয়ে সবসময়েই বদলাচ্ছে।

সিস্টেম	সুজে	স্ল্যাকওয়ার
bin	7.2M	6.2M
boot	6.7M	2.9M
dev	428K	293K
etc	48M	23M
home	114M	3.1M
lib	64M	19M
mnt	2.5K	3.0K
opt	658M	350M
proc	2.0K	16K
root	2.2M	489K
sbin	11M	6.9M
tmp	8.4M	13M
usr	2.5G	1.9G
var	128M	23M
মোট	3.5G	2.3G

লিনাক্স ফাইলসিস্টেম স্ট্যান্ডার্ড অনুযায়ী উপরের ডিরেক্টরিগুলো একটা রুট ডিরেক্টরিতে থাকতে হবে, বা, নিদেন, তাদের লিংক। কিন্তু এগুলো ছাড়াও আরো কিছু ডিরেক্টরি থাকতে পারে। কোনো কোনো ডিস্ট্রোতে এর বাইরেও ডিরেক্টরি থাকে বৈকি, যেমন সুজেতেই '/srv', যেটার মধ্যে সার্ভার সংক্রান্ত তথ্য রাখে। এর মানে, যতটুকুই হোক, গ্নু-লিনাক্স স্ট্যান্ডার্ড থেকে নড়ছে সুজে। এফএসস্ট্যান্ড অনুযায়ী নিচের ডিরেক্টরিগুলো রুট ডিরেক্টরিতে থাকতে পারে।

/initrd — এই '/initrd' ডিরেক্টরি একটা গ্নু-লিনাক্স সিস্টেমকে দেয় বুটলোডার মারফত র‍্যামডিস্ক ব্যবহারের সুযোগ। র‍্যামডিস্ক (RAM Disk) মানে র‍্যান্ডম-অ্যাকসেস-মেমরির একটা বিশেষ এলাকা, র‍্যামডিস্ক বানানোর সফটওয়ার দিয়ে যাকে বানিয়ে নিতে হয়, যে এলাকাটা ঠিক একটা ছোট্ট সাইজের হার্ডডিস্ক পার্টিশনের মত ব্যবহার করবে। হার্ডডিস্ককে নকল বা এমুলেট করবে এই র‍্যামাঞ্চলটা। সুবিধেটা এই যে হার্ডডিস্ক পার্টিশন থেকে তথ্য পড়ার চেয়ে বহুগুণ বেশি গতিতে এইখান থেকে তথ্য পড়ে নিতে পারে সিপিইউ, দ্রুত তার কাজে ঢুকে যেতে পারে। আর যেই মেশিন শাটডাউন বা রিবুট করছি তখনই এই র‍্যামডিস্কটা মুছে যাচ্ছে। একটা সঠিক সাইজের অপারেটিং সিস্টেমকে, র‍্যাম-এলাকাটার মধ্যে যেটা ভরে যাবে, এইভাবে র‍্যামে রেখেই কাজ করে যেতে পারে কম্পিউটার। '/initrd' ডিরেক্টরিটা সুযোগ দেয় র‍্যামডিস্ককে রুট ফাইলসিস্টেম হিশেবে মাউন্ট করার, এরপর এখান থেকেই প্রয়োজনীয় প্রোগ্রাম চালানো যায়। তারপর, প্রাথমিক কাজ শেষ হলে, সঠিক হার্ডডিস্ক পার্টিশনকে রুট ডিরেক্টরি হিশেবে মাউন্ট করে নেওয়া যায়। এই ভাবে র‍্যামডিস্ক রাখার সুযোগ করে দিয়ে '/initrd' ডিরেক্টরি সিস্টেম বুটের গোটা কাজটাকে দুটো অংশে ভেঙে দেয়, প্রথম অংশটায় কারনেলে অনেক কম ড্রাইভার কম্পাইল করা থাকলেও অসুবিধে নেই, দ্বিতীয় স্তরে প্রয়োজনীয় মডিউলগুলো '/initrd' ডিরেক্টরি থেকে তুলে নেয় কারনেল। এই ডিরেক্টরির জটিলতায় আর গিয়ে কাজ নেই, আপনার ইচ্ছে থাকলে আপনি নিজেই শিখে নিতে পারবেন। '/initrd' বানানোর যে কমান্ড, 'mkinitrd', তার ম্যানপেজ পড়ুন। আমাদের এখানে তুলে দেওয়া তালিকাটা দেখুন, সুজে রুট ডিরেক্টরির মধ্যে কোনো '/initrd' ডিরেক্টরি নেই। এই '/initrd' ডিরেক্টরিটা ছাড়াই সুজে কিন্তু র‍্যামডিস্ক ব্যবহার করছে। দেখুন তো, পাঁচ নম্বর দিনে তুলে দেওয়া '/etc/lilo.conf' ফাইলটা থেকে তার কোনো হদিশ পান কিনা।

/home — বারবার ব্যবহার করতে করতে এই ডিরেক্টরিটা আমাদের চেনা হয়ে গেছে। সমস্ত ব্যবহারকারীর নিজের নিজের হোম ডিরেক্টরি বা নিজের ঘর মিলিয়ে তৈরি এই ব্যবহারকারীদের কোয়ার্টার। এই সার্বজনীন ঘরগেরস্থালির মধ্যে একজন বিশেষ ব্যবহারকারী বা ইউজারের হোম ডিরেক্টরির হল '/home/$USER' যার অন্য নাম '~/'। এই দুটোকে কি চিনতে পারলেন? যদি না পারেন, তার মানে আপনি পুরোনো পড়াগুলো ভালো করে তৈরি রাখছেন না। যান, সাত নম্বর দিনের ৯.৩ নম্বর সেকশনটা আর একবার ভালো করে পড়ে আসুন। প্রত্যেকটা ইউজারের নিজের ঘরে কী কী ফাইল থাকে তার সঙ্গে মোটামুটি একটা পরিচয় তো আমাদের আগেই হয়ে গেছে। কিছু ডিরেক্টরি, কিছু ফাইল, আর কিছু সেই সতত-অদৃশ্য ডটানন ফাইল এবং ডিরেক্টরি, যাদের মধ্যে রাখা আছে

বিভিন্ন ধরনের কনফিগারেশন বা সিস্টেম সেটিং। মনে করতে পারছেন সাত নম্বর দিনের ৮.৪ নম্বর সেকশনের আলোচনা থেকে সেই '/etc/skel' ডিরেক্টরির কথা যার স্কেলিটনের বা কঙ্কালের আদলে একজন নতুন ব্যবহারকারীর নতুন হোম ডিরেক্টরি তৈরি করে নেওয়া হয়? এই '/home' ডিরেক্টরিটা, ব্যবহারকারীদের কল্যাণে, বড্ড দ্রুত ভরে যেতে শুরু করে। অনেকসময় তাই এই ডিরেক্টরিটাকে আলাদা একটা পার্টিশনে দিয়ে দেওয়া ভালো, যাতে বেড়ে গিয়ে গিয়ে মূল রুট পার্টিশনের ফাইলগুলোরই দমবন্ধ করে না-দেয়।

/lost+found — দেখুন তো, এই মহাপ্রভুর নামটা আপনার চেনা চেনা লাগছে, কোথায় যেন শুনেছিলেন? ইএক্সটিটু এবং ইএক্সটিথ্রি ফাইলসিস্টেমের ক্র্যাশ-রিকভারি সংক্রান্ত আলোচনা মনে পড়ছে? আমি আগে এটার কথা জানতাম না। তখনো অব্দি আমি ম্যানড্রেক ব্যবহার করছিলাম, একদিন শখ হল, একটু রেডহ্যাট ইনস্টল করে দেখি। প্রথমত, ইনস্টল করতে করতে এবং বুট করতে করতে চোখে প্রায় জল এসে গেছে, তার আগে অব্দি ম্যানড্রেকে এক্সএফএসের ওই দুর্ধর্ষ গতিতে কাজ করে এসেছি, মনে হচ্ছে সবকিছু যেন স্লো-মোশনে ঘটছে, এরপর ইনস্টল হওয়া সিস্টেমে লগ-ইন করে '`ls`' মেরে দেখি, '/lost+found' জাতীয় মেলো-রোমান্টিক নাম, সুখেন দাস কোথায় লাগে? না, আসলে ব্যাপারটা ততটা মিস্টি নয়, ক্র্যাশ-রিকভারির সময় ফাইলসিস্টেম চেক করে পুনরাহরিত ঘেঁটে যাওয়া তথ্য রাখা থাকে এই ডিরেক্টরিতে, ইএক্সটিটু ফাইলসিস্টেমে। অচিরেই রাইজার ফাইলসিস্টেমে রেডহ্যাট ইনস্টল করে আমি মুক্তি পেয়েছিলাম এর থেকে, তার কিছুদিন বাদে রেডহ্যাটটাই উড়িয়ে দিয়েছিলাম। আপনি যদি স্লোমোশন ভালোবাসেন তাহলে ইএক্সটিতে রেডহ্যাট করুন, এবং এই ডিরেক্টরির শোভা দেখুন বসে বসে, কারণ অনেকটা সময়ই তো আপনাকে বসে থাকতে হবে, প্রতিবার কমান্ড দিয়ে প্রম্পট ফেরত আসার জন্যে।

/proc — এই ডিরেক্টরির গোটা ফাইলসিস্টেমটাই অন্য প্রতিটি ডিরেক্টরির থেকে একদম আলাদা এই অর্থে যে এটা কোনো বাস্তব ফাইলব্যবস্থাই না, একটা বিমূর্ত এবং সাঙ্কেতিক ফাইলব্যবস্থা। সিস্টেমের প্রসেসগুলোর বিষয়ে তথ্য যোগানোর উদ্দেশ্যে বানানো একটা নকল ফাইলব্যবস্থা। আমরা, কী কী ফাইলব্যবস্থা আমাদের সিস্টেমে ব্যবহার হচ্ছে সেটা জানার জন্যে, '`cat /proc/filesystems`' কমান্ড ব্যবহার করেছিলাম, মনে আছে? কোনো বাস্তব ফাইল '/proc' ডিরেক্টরিতে থাকেনা। থাকে শুধু সিস্টেম কাজ করার সময়ে সিস্টেমের মেমরি, সিস্টেমে কী কী ডিভাইস মাউন্টেড আছে, সিস্টেমের হার্ডওয়ার কনফিগারেশন ইত্যাদি সিস্টেম-তথ্য। '/proc' ডিরেক্টরিতে গিয়ে, এই গোটা পাঠমালা জুড়ে বাস্তব ফাইল নিয়ে আপনি যা যা জেনেছেন সেটা দিয়ে কদাচ বোঝার চেষ্টা করবেন না। যেমন, আপনি জানেন একটা ফাইলে টেক্সট থাকার সঙ্গে ফাইলের বাইটমাপের সম্পর্ক। এবার '`cat /proc/filesystems`' করে স্ক্রিনে চিহ্নের সংখ্যা গুনে নিয়ে আপনি যদি সেটা দিয়ে '/proc/filesystems' ফাইলটার সাইজের সঙ্গে মেলাতে যান, প্রচণ্ড চমকাবেন। আপনি '`ls -sh`' করে দেখুন গোটা '/proc' ডিরেক্টরির যাবতীয় ফাইলেরই সাইজ দেখবেন শূন্য, 'kcore', 'mtrr', আর 'self' বাদ দিয়ে। এবং মজার কথা কী বলুন তো, আপনার অত্যাবশ্যক পার্টিশনগুলো ছাড়া আর কিছু যখন মাউন্ট করা নেই, মানে আমাদের মাউন্টের ছবিতে স্টপককহীন ওই তিনটে চৌবাচ্চা বাদ দিয়ে আর কাউকেই যখন সিস্টেম 'তুমি যে আমার' করেনি, তখন আপনি '`du -chs /proc`' করে যে সাইজ পাবেন সেটা নাটকীয়ভাবে বদলে যাবে, যেই আপনি পার্টিশনগুলো মাউন্ট করে নেবেন। আমার মেশিনের সুজে সিস্টেমে যখন স্ল্যাকওয়ার, আর্কাইভ আর উইনডোজ পার্টিশনগুলো মাউন্ট করা নেই তখন '/proc' ডিরেক্টরির 'du' মানে ডিস্ক-ইউসেজ বা ডিস্ক-ব্যবহার দেখাচ্ছে চুরানব্বই কিলোবাইট। আর ওগুলো মাউন্ট করা মাত্রই ডিস্ক-ব্যবহার বেড়ে গিয়ে হচ্ছে দুশোপঁচিশ মেগাবাইট। এবং তখনও যাবতীয় ফাইলের আকার শূন্য, একা 'kcore' ফাইলটাই প্রায় ওই গোটা সাইজটা নিয়ে নিচ্ছে। কিন্তু আর না, আমরা আমাদের পাঠমালার এক্তিয়ার ছাড়িয়ে বেরিয়ে যাচ্ছি।

/root — এটা হল রুট ইউজারের বা সিস্টেম অ্যাডমিনিসট্রেটরের নিজের হোম ডিরেক্টরি। '/' বা রুট ডিরেক্টরির মধ্যে এই রুট বা '/root' নামের ডিরেক্টরিতে গুলিয়ে ফেলবেন না। অনেক আগে ওই '/' ডিরেক্টরিটাই ছিল রুট ব্যবহারকারীর হোম-ডিরেক্টরি। সেখানেই থাকত তার নিজের ফাইলপত্তর। পরে, ঘরকন্নার কাজ একটু পরিপাটী রাখার উদ্দেশ্যে এই '/root' ডিরেক্টরিটা বানানো হয়। এই ডিরেক্টরিটা যেহেতু একজন ব্যবহারকারীর হোম ডিরেক্টরি, একে '/home' ডিরেক্টরির মধ্যেই আর একটা সাবডিরেক্টরি, '/home/root' নামেই তো রাখা যেত?

সেটা করা হয়না, তার কারণ, আমাদের মত পিসি ব্যবহারকারীদের জন্যেই শুধু গ্নু-লিনাক্স তৈরি হয়নি। নেটাবদ্ধ বড় বড় সিস্টেমে অনেক সময় ভৌগোলিক দূরত্বে থাকা একটা পার্টিশনকেও মাউন্ট করা হয় '/home' ডিরেক্টরিতে। সেটাও তখন স্টপকক লাগানো চৌবাচ্চা হয়ে যায়। কিন্তু অন্য ব্যবহারকারীদের সাময়িকভাবে বাইরে রেখেও একটা সিস্টেম চালু হলেও, সুপারইউজারকে তো মূল সিস্টেমের অত্যাবশ্যকীয় অংশের মধ্যে থাকতেই হবে। তাই এই ব্যবস্থা। এই সবিশেষ ব্যবহারকারীর একটা আলাদা সুপারভাইজর কোয়ার্টার একদম রুট পার্টিশনের মধ্যেই।

গ্নু-লিনাক্স ফাইলসিস্টেম হায়েরার্কির প্রাথমিক ধারণাটা হল আমাদের। রুট ডিরেক্টরির মধ্যে কী কী ডিরেক্টরি থাকে সেটা আমরা জানলাম। এবার এই ডিরেক্টরিগুলোর চারটেকে নিয়ে আরো দু-চারটে কথা বলার আছে। আলোচনার সূত্রেই বলেছিলাম আমরা, '/bin', '/etc', '/sbin' এবং '/usr' — এই চারটে ডিরেক্টরি আলাদা করে আরো একটু বলে নিতে হবে। সেটা দিয়ে আমরা বরং সামনের দিনের আলোচনা শুরু করব। সেটাই আমাদের এই জিএলটি ইশকুল পাঠমালার শেষ দিন।

আজ চোদ্দই জানুয়ারি। একটু বেশি সময় লাগল এটা শেষ হতে। তবে, গত দুদিন গায়ে জ্বর থাকায় বসে থেকেছি হয়ত, কিন্তু লিখতে তেমন পারিইনি। আর কলেজে পরীক্ষা চলছে, তাই যেতেই হচ্ছে। আপনাদের পড়তে পড়তে কতটা হচ্ছে জানিনা, আমার লিখতে লিখতে ইতিমধ্যেই ক্লান্ত লাগছে। এখনো পর্যন্ত শব্দসংখ্যা একলাখ সতেরো হাজার। আর পুরোনো ফাইলগুলো থেকে দেখছিলাম, জিএলটি পাঠমালার এই ফাইনাল ভার্শনটায় হাত দিয়েছি পয়লা নভেম্বরের আগে না, তার মানে, পঁচাত্তর দিনে একলাখ সতেরো হাজার শব্দ লিখেছি, গড়ে প্রত্যেকদিন ১৫৬০-খানা করে শব্দ লিখেছি, এর মধ্যে ব্রেক গেছে একদিন দুপুর, সাত ঘন্টা, আর একদিন রাত্তিরে তিন ঘন্টা। এছাড়া কলেজ যাওয়ার সময়টুকু বাদ গেছে। সে তুলনায় হয়ত কম লাগছে, কিন্তু তার সঙ্গে যোগ করুন তো মোট ওই অতগুলো ছবি। আর এই লিখতে লিখতেও কী পরিমাণ যে আমায় পড়তে হয়েছে সে আর বলার নয়। সত্যিই বড্ড খাটনি। এটা শেষ হলে গ্নু-লিনাক্স কমিউনিটির কাছে আমার ঋণশোধ, পেয়িং ব্যাক টু দি কমিউনিটি, বোধহয় বেশ ভালোভাবেই হবে। আর মাত্র এক দিন। লে বংকা।

glt-mad@ilug-cal.org

MADHYAMGRAM

সংকলন ও রচনা : মধ্যমগ্রাম জিএলটি-র (glt-mad@ilug-cal.org) তরফে : ত্রিদিব সেনগুপ্ত

গ্নু-লিনাক্স ইশকুল

জিএলটি মধ্যমগ্রাম
glt-mad@ilug-cal.org

আপনাদের কার কী প্রতিক্রিয়া জানিনা, অশেষ আট নম্বর দিনের, মানে আমাদের এই সিরিজের নয় নম্বর চ্যাপ্টারের লেখা পড়ে যা বলেছে সেটা অনেকটা এরকম, বাঁশ ক্রমে আসিতেছে, আমরা প্রাকৃতজনেরা বিবিধ উষ্ণতায় চিহ্নিত হইব, রক্তিমতা প্রভাব বিস্তার করিবে — ইত্যাদি। মানে, পড়া এবং মনে-রাখাটা এতটাই বেদনার হয়ে যাচ্ছে। তবে, তার একটা বড় কারণ, যখন যখন যা যা করতে বলা হয়েছে, সেগুলো যথেষ্ট রকমে ও করেনি, গোটাটা তাই মাথা থেকে বেরিয়ে যাচ্ছে হয়ত। এটা করতেই হবে, ব্যাশ শেলে যেমন বাঁ-হাতের কড়ে আঙুলটা ট্যাবের দিকে চলে যাওয়ার অভ্যস্ততা, তেমনি, প্রাত্যহিক কমান্ডগুলো তাদের অত্যাবশ্যক অপশান সহ, পুরো প্রতিবর্তী ক্রিয়ায় চলে আসা চাই। প্রতি মুহূর্তে সিস্টেমের ভিতরে আপনার নড়াচড়ায়। আগের দিনের বকেয়া ফাইলসিস্টেম হায়েরার্কির আলোচনাটুকু সেরে নিয়েই, আজ, এই পাঠমালার শেষ দিনে, গ্নু-লিনাক্স সিস্টেমের মধ্যে আমাদের নড়াচড়াটাকেই আমরা বাড়িয়ে তুলব।

।। দিন নয়।।

১।। বকেয়া হায়েরার্কি

রুট ডিরেক্টরি '/'-এর ভিতরে চারটে সাবডিরেক্টরি '/bin', '/etc', '/sbin', এবং '/usr' — এদের নিয়ে আলোচনাটুকু আমাদের বাকি আছে আগের আট নম্বর দিনের থেকে। এই চারটেকে আলাদা করে একটু বিশদ করে কথা বলার দরকার আছে। এর মধ্যে '/bin', এবং '/etc' ডিরেক্টরিদুটোর সঙ্গে একটা হালকা মোলাকাত আমাদের আগেও হয়েছে। সেটাকে এবার আমরা বাড়িয়ে তুলব। একটা কথা, অন্য অনেক ডিরেক্টরিতেই অনেক প্রয়োজনীয় খুঁটিনাটি আছে, যা বুড়ি-ছোঁয়া করে বেরিয়ে এসেছি, যেমন '/lib' বা '/initrd', তার একটা বড় কারণ এই যে, এই পাঠমালাটার প্রোজেক্ট হল আপনাকে গ্নু-লিনাক্স সিস্টেম বুঝতে এবং ভাবতে শুরু করায় সাহায্য করা, কিন্তু তা দিয়ে এগোনো যাবে খুব সামান্যই। নিজে সাঁতার কাটা তো ছেড়ে দিন, হাত-পা ভালো করে ছুঁড়তে হলেও, এই জানাগুলোকে অনেকটা বাড়িয়ে তুলতে হবে। আমি চেষ্টা করেছি সেই জানার প্রক্রিয়ার লগিগুলো গাছের গোড়ায় গোড়ায় রেখে আসতে, কতটা সেটা কাজের হয়েছে তা আপনিই বুঝবেন। আর পারলে, একটু মেল করবেন, মদনার মত বসে থাকবেন না, পড়ার পর। পরে, আমাদের অনেকেরই সেটা কাজে লাগবে। এই পারস্পরিকতাগুলোর উপরেই গ্নু-লিনাক্স, তার সফটওয়ার, তার ডকুমেন্টেশন, সব কিছু বেঁচে থাকে।

এই চারটে ডিরেক্টরির ভিতর '/bin' এবং '/sbin' — এই দুটোর একটা মিল আছে, নামেও দেখুন, দুটোতেই থাকে বাইনারি ফাইল। বাইনারি ফাইল কাকে বলে মনে আছে? হাইলেভেল বা উচ্চস্তরের ভাষায়, মানে মানববোধ্য কোনো কম্পিউটার ভাষায় লেখা মূল সোর্স কোডকে কম্পাইল করে, বা মেশিনবোধ্য করে তুলে, যে এক্সিকিউটেবল বা চালনীয় প্রোগ্রাম তৈরি হয়, সেটাই হল বাইনারি। এই বাইনারিদেরও, ঠিক আমাদের মতই, জাত আর বেজাত আছে। সিস্টেম বাইনারিরা থাকে বিশেষ জায়গায়, এসবিনে, রুট পাসওয়ার্ড ছাড়া যেখানে হাত দেওয়া যায়না, তাদের চালাতে গেলেও লাগে সেই রুট পাসওয়ার্ড। আর বিনে থাকে আমজনতার আমবাইনারিরা।

১.১।। '/bin' ডিরেক্টরি

এই '/bin' ডিরেক্টরিতে থাকে অনেকগুলো অত্যাবশ্যক কমান্ড বা আদেশ। এদের রুট বা সাধারণ ব্যবহারকারী দুজনেই ব্যবহার করতে পারে। প্রতিমুহূর্তে আমাদের আর সিস্টেমের মধ্যে দোভাষির কাজ করে চলেছে যে শেল, গ্নু-লিনাক্স সিস্টেমে সচরাচর শেল বলতে ব্যাশ শেলই ব্যবহার করা হয়, যদিও অন্য শেলও চাইলে করা যায়, সেই শেলও থাকে এই এই '/bin' ডিরেক্টরিতেই। এছাড়া কিছু বাইনারি থাকে '/usr/bin' এবং '/usr/local/bin' ডিরেক্টরিতেও, তবে তাদের তুলনায় এই '/bin' ডিরেক্টরির বাইনারিরা প্রাত্যহিক কাজের বেলায় অনেক বেশি জরুরি। আমরা যে প্রোগ্রামগুলো পরে সিস্টেমে যোগ করে নিতে পারি, যেমন মাল্টিমিডিয়া ফাইল উপভোগের জন্যে এই ব্রহ্মাণ্ডের শ্রেষ্ঠ প্রোগ্রাম 'mplayer', অন্যান্য ব্রহ্মাণ্ডে অবশ্য আরো ভালো প্রোগ্রাম আছে বলে শুনেছি, শব্দব্রহ্মকে ধরে রাখা এবং বিভিন্ন অবতারে তাকে প্রকাশ করার জন্যে 'lame', বা, শুধু গান-বাজনা-ফিলিম দেখলে লোকে কী বলে, মাঝে

মাঝে কিছু লেখাপড়াও তো করতে হয়, তার জন্যে একটা অনবদ্য ডিকশনারি ‘wn’ ইত্যাদি — এদের অনেকেই থাকে ‘/usr/local/bin’ ডিরেক্টরিতে। তবে চাইলে তাদের অন্য জায়গাতেও রাখা যায়। এই প্রোগ্রামগুলো সে অর্থে ঐচ্ছিক, অত্যাবশ্যক নয়। এমনকি শুধু ‘/’ ডিরেক্টরিটুকুই যদি মাউন্ট করা হয়, তখনো যে কমান্ডগুলোকে লাগবেই ন্যূনতম কাজটুকু করতে গেলে, তারা থাকে এই ‘/bin’ ডিরেক্টরিতে। ধরুন আপনার সিস্টেমে কোনো কেলো ঘটেছে এবং আপনি সেটা মেরামত করার চেষ্টা করছেন, তখনো আপনার এই বিন ডিরেক্টরির বাইনারিদের লাগবেই। সিস্টেমে যে বুট-স্ক্রিপ্ট গুলো থাকে, মানে বুট করার সময়ে কিছু বিশেষ প্রোগ্রাম চালিয়ে নেওয়ার দরকার পড়ে, তারাও থাকে এই বিন ডিরেক্টরিতে। আমরা যে গ্নু-লিনাক্স এফএসস্ট্যান্ড বা ফাইলসিস্টেম স্ট্যান্ডার্ডের কথা বলেছিলাম, সেই স্ট্যান্ডার্ড অনুযায়ী যে যে কমান্ড বা তাদের লিংক এই ‘/bin’ ডিরেক্টরিতে থাকতেই হবে, তাদের প্রায় গোটা তালিকাটা এখানে তুলে দিই। এদের মধ্যে বেশ কিছু কাপ্তেনকে আপনি ইতিমধ্যেই চেনেন। খুব ছোট একটা পরিচিতিও দিচ্ছি, ভালো করে আপনি পড়ে নিন ম্যানুয়াল থেকে।

cat	ফাইলদের কনক্যাটেনেট করে	ls	ডিরেক্টরির অন্তর্বস্তু দেখায়
chgrp	ফাইলের গ্রুপ-মালিকানা বদলায়	mkdir	নতুন ডিরেক্টরি বানায়
chmod	ফাইল-ব্যবহারের অনুমতি বদলায়	more	পাতার এককে টেক্সট দেখায়, লেসের মত
chown	ফাইলের মালিকানা বদলায়	mount	পার্টিশনে মাউন্ট করে
cp	কপি করে	mv	ফাইলের স্থানান্তর বা নামান্তর করে
date	সময় ও তারিখ দেখায়	ps	চলমান প্রসেসদের তালিকা দেখায়
dd	ফাইলের ফরম্যাট বদলায় ও কপি করে	pwd	বর্তমান কাজের ডিরেক্টরির নাম দেখায়
df	ডিস্কভূমির ব্যবহারের তালিকা দেয়	rm	ফাইল বা ডিরেক্টরি ওড়ায়
echo	প্রদত্ত টেক্সট ফুটিয়ে তোলে	rmdir	ফাঁকা ডিরেক্টরি ওড়ায়
hostname	সিস্টেমের নাম ধার্য করে	sh	বর্ন শেল চালু করে
kill	প্রসেসদের নানা সিগনাল পাঠায়	su	ইউজার পরিচিতি বদলায়
ln	ফাইলের লিংক তৈরি করে	sync	বাফার ফাঁকা করে
login	সিস্টেমে একজন ইউজারকে ঢুকতে দেয়	umount	পার্টিশন আনমাউন্ট করে

ফাইলসিস্টেম স্ট্যান্ডার্ড-এর মোট তেত্রিশটা আদেশের মধ্যে একদম অপরিচিত সাতটা কমান্ড আমরা বাদ দিয়েছি। এই ছাব্বিশটা কমান্ড প্রত্যেকটাই আমাদের আলোচনায় এসেছে। দু-একটা একটু মনে পড়িয়ে দিই, ‘ps’ এসেছিল আমাদের অগোচরে একটা সিস্টেমে কত কত প্রসেস একসঙ্গে চলতে থাকে তার আলোচনায়, আর ‘sync’ এসেছিল একটা পার্টিশনে কী কী ব্লক থাকে, তাদের কারনেল কী ভাবে কাজের আগে মেমরিতে বা সোয়াপ ফাইলে তুলে নেয়, এবং কাজের শেষে ফের পার্টিশনে লিখে দেয় তার আলোচনায়।

‘hostname’ কমান্ডটা যে কোনো গ্নু-লিনাক্স সিস্টেমের একটা বৈশিষ্ট্যকে চিহ্নিত করে। একটা গ্নু-লিনাক্স সিস্টেম সবসময়েই, সংজ্ঞাগতভাবেই, একটা নেটওয়ার্কাবদ্ধ সিস্টেম। এমনকি সেটা যদি মাত্র একটা মেশিনেই চলে, তখন সেটা এক মেশিনের নেটওয়ার্ক। এই ধারণাটা গ্নু-লিনাক্স চিন্তাপদ্ধতির একদম আভ্যন্তরীন। সবকিছুই একটা সার্ভারের ধারণা নিয়ে চলে। পুরো জিইউআই বা গুইটা হল একটা এক্স-সার্ভার, মেল দেওয়া-নেওয়াটা হল মেল সার্ভার, ইত্যাদি। যখন আপনি আলাদা করে আপনার মেশিনের কোনো নাম দেননি, তখন আপনার মেশিনের ডিফল্ট নামটা হল ‘localhost’, আর আপনার নেটওয়ার্কের ডিফল্ট নাম হল ‘localdomain’। এই ‘localdomain’ ডোমেইনের এই মেশিনটার নিজস্ব ঠিকানা তখন ‘localhost.localdomain’, মধ্যের বিন্দু বা ডটটা বা ‘.’ দিয়ে মেশিনটার ঠিকানা বোঝানো হচ্ছে। ধরুন, আমার মেশিনের নাম ‘mahammad’, তাই সেটার পুরো ঠিকানা হল ‘mahammad.localdomain’। এই নেটওয়ার্কে অন্য কোনো মেশিন যোগ হলে সে এই মেশিনটায় কোনো ফাইল লিখতে চাইলে, বা এখান থেকে কোনো ফাইল পড়তে চাইলে এই ঠিকানাটা ব্যবহার করবে। মেশিনের এই নামটাকে আমি বদলাতে পারি ‘hostname’ কমান্ড দিয়ে।

‘more’ কমান্ডটার আলোচনা আমরা ‘less’ কমান্ডের আলোচনার সূত্রেই করেছি। আর একটা পেজার, পেজ বাই পেজ, কোনো একটা টেক্সটকে স্ক্রিনে ফুটিয়ে তোলে। আর এই ‘kill’ কমান্ডটা খুবই কাজের জিনিষ, কখনো কোনো

একটা প্রসেস কোনো ঝামেলা পাকালে খুব কাজে আসে। নানা কিছু করা যায় একটা প্রসেসকে নিয়ে, নানা ধরনের হত্যা, শ্বাসরোধ, বা ছুরিকাঘাত থেকে শুরু করে কাতুকুতু দিয়ে মারা অব্দি। ভালো করে ম্যানপেজ করে তবে কিলার হবেন, কিলার হো তো স্টোনম্যান জ্যায়সা, এখনো কোনো হদিশ হলনা। ফুটপাথবাসী গরিব প্রসেসদের মারলেন তাতে তেমন কোনো ব্যথা নেই, কিন্তু এই করতে গিয়ে যদি কোনো সিস্টেম ডিমন বা যখ গোছের কোনো প্রসেসের গায়ে হাত পড়ে যায়, তাহলেই সমূহ কেলেংকারি। তাই, ... *ম্যান পড়ার নেশা করো ম্যান*

আর এই 'sh' কমান্ডটা খেয়াল করুন, এটা হল মূলত বর্ন (Bourne) শেলের কমান্ড, বেল ল্যাবরেটরির ইউনিক্সের সাত নম্বর ভার্শন থেকে শুরু হয়েছিল, গ্নু-লিনাক্সের মূল শেল ব্যাশের নামকরণ হয়েছিল এই বর্ন শব্দটার সঙ্গে মিলিয়ে, বর্ন-এগেন (**B**ourne-**A**gain-**SH**ell) বা ব্যাশ (bash)। এই 'sh' কমান্ডটা আসলে একটা লিংক, এটা ডেকে আনে ব্যাশ শেলকেই। শুধু, যখন সরাসরি 'bash' কমান্ড না-দিয়ে 'sh' কমান্ড দিয়ে তাকে ডাকা হয় ব্যাশ শেল যদ্দূর সম্ভব নিখুঁতভাবে তার পূর্বপুরুষ ওই বর্ন-শেলের মত আচার আচরণ করার চেষ্টা করে। ভালো করে বুঝতে হলে ম্যান পেজ পড়ুন। ধরুন আমি এই ব্যাশ শেলের ম্যানপেজটাকে স্ক্রিনে পড়তে চাইছি, বা প্রিন্ট নিতে চাইছি। এবার, আমি এর এমন একটা প্রতিরূপ বানাতে চাইছি যা আমি আমার কলেজের কম্পিউটারে, সেখানে লোকজন কেউ তেমন সিরিয়াসলি কম্পিউটার ব্যবহার করেনা, শুধু উইনডোজই আছে, আমি সেখানে বসেও ফাইলটা পড়তে চাই। প্রথমে ম্যান করে পাওয়া ব্যাশের ম্যানুয়াল পাতাটাকে রিডাইরেক্ট করুন একটা ফাইলে, '`man bash > bashman`'। তারপর ফাইলটাকে এইচটিএমএল করে নিন, '`rman -f html bashman > bashman.html`' কমান্ডে, 'rman' দিয়ে। এবার, এই 'bashman.html' ওয়েবপেজটা সরাসরি রাখতে পারেন, ব্রাউজার তো একটা যে কোনো সিস্টেমেই থাকবে, বা পিডিএফ করে নিতে পারেন, হুবহু যেমন ফরমাটে আপনি টেক্সটটা পড়তে চান সেরকমই করে রাখতে পারেন, তাতে যে কোনো মেশিন থেকে অবিকল একই চেহারায় পড়তে বা প্রিন্ট নিতে পারবেন। সেটার জন্যে পরপর দুটো কমান্ড কাজে লাগাতে হবে ।'`html2ps bashman.html > bashman.ps`', এতে ও একটা গড় ডিফল্ট ফরম্যাটে পোস্টস্ক্রিপ্ট বানিয়ে দেবে, যার নাম 'bashman.ps'। এর সঙ্গে ম্যানপেজ পড়ে দেখুন, কত কিছু আপনি করতে পারেন, কোন ফন্ট হবে, সূচীপত্র থাকবে কিনা, এতে ছবি দেবেন কিনা, হেডার ফুটার কী হবে, পাতার সাইজ কী হবে, সবই আলাদা করে দিয়ে যাওয়া যায় অপশান দিয়ে। এবার, পিডিএফ পাওয়ার জন্যে কমান্ড দেবেন, '`ps2pdf bashman.ps`', যাতে সেই পোস্টস্ক্রিপ্ট থেকে পিডিএফ হবে। এখানে আর রিডাইরেক্ট করতেও হবেনা, ও নিজেই নতুন ফাইলটার নাম করে নেবে '`bashman. pdf`', সেরকমই ব্যবস্থা করা আছে, ম্যানপেজ পড়ে দেখুন। এখানে আমরা চারটে কমান্ড কাজে লাগালাম, 'man', 'rman', 'html2ps', 'ps2pdf'। আরো বহুভাবে করা যায়। আমি সচরাচর করি 'htmldoc' দিয়ে এই কাজটা করি, কিন্তু সেটা সব ডিস্ট্রোয় ডিফল্টে থাকেনা, আলাদা করে ইনস্টল করে নিতে হয়, একদমই না-থাকলে আপনি তো ডাউনলোড করেই ইনস্টল করে নিতে পারবেন। 'html2ps' আর 'ps2pdf' মোটামুটি থাকেই। নাম দেখে আন্দাজ করুন, প্রথমটা ওয়েবপেজ থেকে পোস্টস্ক্রিপ্ট বা পিএস ফাইল বানায়, আর দ্বিতীয়টা সেই পিএস থেকে পোর্টেবল ডকুমেন্ট ফরম্যাট বা পিডিএফ বানায়। এই পিডিএফটা সত্যিই বানিয়ে নিন — আজকের আলোচনায় আমাদের মূল জায়গাটা জুড়েই চলবে ব্যাশ, তার জন্যে বারবার দেখা দরকার পড়বে এই ডকুমেন্টটা, পারলে একটা প্রিন্ট-আউট বার করে নিন।

যদি আপনি ব্যাশ শেল ছাড়া অন্য কোনো শেলও ইনস্টল করে থাকেন, ব্যবহার করতে চান, তাহলে সেই শেলের প্রোগ্রাম ফাইলটা, বা তার লিংকটা থাকবে এই '/bin' ডিরেক্টরিতে, যাতে আপনি কোনো কমান্ড দিলে শেল তার পথনির্দেশে দেওয়া ডিরেক্টরিগুলো খুঁজেই এটা পেয়ে যেতে পারে। সুজে আর স্ল্যাকওয়ার দুটো ডিস্ট্রোতেই '/bin' ডিরেক্টরিতে 'bash' আর 'sh' দুটোই আছে। যে নামেই তাকে ডাকুন, তাঁর করুণাটা একই থাকবে। পাঁচ নম্বর দিনের শেলের পথনির্দেশটা দেখে নিন। এর মধ্যে অনেকগুলো ডিরেক্টরি আছে। এর মধ্যে একাধিক অংশই শেষ হচ্ছে '/bin' দিয়ে। রুট ডিরেক্টরির '/bin' ডিরেক্টরিতে অত্যাবশ্যকীয় কমান্ডগুলোর বাইরে অন্য কমান্ডরা থাকে এই বাড়তি '/bin' ডিরেক্টরিগুলোয়। আপনার ডিস্ট্রো নিজেই যাদের ইনস্টল করে আপনার সিস্টেমে, বা, পরে, বাইরে থেকে, আপনি যে বাড়তি প্রোগ্রামগুলো ইনস্টল করে নিতে পারেন, কিছু দেওয়া থাকে আপনার ইনস্টলেশন সিডিতে বা ডিভিডিতেই, এরও বাইরে কোনো বহিরাগত প্রোগ্রামকেও আপনি ইনস্টল করে নিতে পারেন, যেমন আগেই বলেছি — তারা সবাই থাকে এই অন্যান্য '/bin' গুলোয়। মনে করতে পারছেন পাঁচ নম্বর দিনের ১১ নম্বর সেকশনে ইউজার

‘dd’-র শেলের পথনির্দেশটা, ‘/home/dd/bin:/usr/local/bin:/usr/bin:/usr/X11R6/bin:/bin:/usr/games:’ যা ‘`echo $PATH`’ কমান্ড দিয়ে পেয়েছিলাম? এর মধ্যে দেখুন, রুট ডিরেক্টরির ‘/bin’ ছাড়াও, আরো চারটে ‘/bin’ আছে — ‘/home/dd/bin’, ‘/usr/local/bin’, ‘/usr/bin’ এবং ‘/usr/X11R6/bin’। তবে, আগে দেওয়া ওই অত্যাবশ্যকীয় কমান্ডগুলোর বাইরেও, আরো কয়েকটা প্রোগ্রাম, বা তাদের লিংক, যদি আপনি প্রোগ্রামগুলো আপনার সিস্টেমে ইনস্টল করেন, থাকার কথা মূল ‘/bin’ ডিরেক্টরিতেই। এরা প্রত্যেকেই কিন্তু সেই অর্থে ঐচ্ছিক বা অপশনাল।

`csh`	ব্যাশের মতই আর একটা শেল, এর নাম ‘C’
`ed`	এই কমান্ড এডিটরটার কথা আমরা আগে একাধিকবার উল্লেখ করেছি
`tar`	ফাইল বা ডিরেক্টরি বা তাদের সমাহারকে একটা সিন্দুক ফাইলে পরিণত করে, মনে করুন
`cpio`	‘tar’-এর মতই আর একটা আর্কাইভ বা সিন্দুক বানানোর উপযোগিতা-সফটওয়ার
`gzip`	গ্নু-র বানানো এই কোঁকড়ানোর বা কমপ্রেস করার কমান্ডটাও আমরা আগেই ব্যবহার করেছি
`gunzip`	‘gzip’ দিয়ে কোঁকড়ানো ফাইলকে ফের স্বাভাবিক করে
`zcat`	কোঁকড়ানো ফাইলকে ‘cat’ করে
`netstat`	মেশিনে সংযুক্ত নেটওয়ার্ক সংক্রান্ত তথ্য দেয়
`ping`	নেটওয়ার্ক পরখ করার সফটওয়ার

এই উপরের প্রোগ্রামগুলো বা কমকরে তাদের লিংক ‘/bin’ ডিরেক্টরিতে রাখার মানে কী — একটু ভাবুন তো, যেমন বলেছিলাম, সিস্টেমটা যারা বানিয়েছে তাদের চিন্তাকে চেনার চেষ্টা করুন। শুধু একটা কথা মনে পড়ান, গ্নু-লিনাক্স প্রথম থেকেই ধরে নেয় আপনার মেশিনটা একটা নেটওয়ার্কে আবদ্ধ আছে অন্য এক বা একাধিক মেশিনের সঙ্গে। ‘tar’, ‘cpio’, ‘gzip’ আর ‘gunzip’ প্রোগ্রামগুলোর প্রয়োজনীয়তা সহজেই বোঝা যায়, সিস্টেম ঘেঁটে গেলে ব্যাকআপ থেকে ফের বানিয়ে নেওয়ার সময়ে কাজে লাগবে। সেই একই কাজে ‘zcat’। কম্পিউটার থেকে কম্পিউটারে প্রয়োজনের বদলে এই ‘/bin’ ডিরেক্টরিতে কী কী ফাইল থাকবে তার নিরিখটাও বদলায়।

১.২।। ‘/sbin’ ডিরেক্টরি

পাঁচ নম্বর দিনের ১১ নম্বর সেকশনে দেওয়া ইউজার ‘dd’-র শেলের পথনির্দেশ তো দেখলাম আমরা, সেখানে সর্বশক্তিমান রুটের শেলেরও পথনির্দেশটা দেওয়া ছিল, যার শুরুতেই ছিল ‘/sbin’, তার পরে কয়েকটা ডিরেক্টরি ছিল, ‘/usr/sbin’, ‘/usr/local/sbin’, ‘/root/bin’, ‘/usr/local/bin’, ‘/usr/bin’, ‘/usr/X11R6/bin’, ‘/bin’, ইত্যাদি, মনে পড়ছে? পরের ‘/bin’ অংশগুলো তো আমরা চিনি, আগের সেকশনেই দেখলাম এইমাত্র, যেসব কমান্ড একজন সাধারণ ইউজার চালাতে পারে, তার সবগুলোই রুটও পারে। কিন্তু আরো কিছু কমান্ড চালাতে পারে রুট, সিস্টেম কমান্ড, যা ‘dd’ পারেনা, সেই কমান্ডগুলো থাকে ওই ‘/sbin’ ডিরেক্টরিগুলোয়। প্রথমটা একদম রুট ডিরেক্টরি মানে ‘/’ বা স্ল্যাশ-এ, পরেরগুলো স্ল্যাশের নানা সাবডিরেক্টরিতে। জাতের ভিত্তিতে অধিকারভেদের এমন নিপাট এবং কঠোর ব্যবস্থা আরএসএস বা মুসলিম লিগেরও ধারণার বাইরে। সর্বময় ‘root’ শুধু যাদের চালাতে পারে সেই অত্যাবশ্যক সিস্টেম কমান্ডগুলো থাকে ‘/sbin’ ডিরেক্টরিতে। আট নম্বর দিনের ৩ নম্বর সেকশন থেকে, সিস্টেম চালু বা বুট হওয়ার সময় এবং বন্ধ বা শাটডাউন হওয়ার সময় পার্টিশনগুলো একের পর এক কী ভাবে মাউন্ট বা আনমাউন্ট হয়, সেই গোটা প্রক্রিয়াটা, তাতে কী কী প্রোগ্রাম কাজে লাগায় সিস্টেম, সেটা মনে পড়ছে? এদের রুট পার্টিশন স্ল্যাশের মধ্যেই রাখা হয় ওই প্রথমতম পার্টিশনের কথা ভেবে, বুট করার সময় এই ডিরেক্টরিটা তো প্রথম মাউন্ট হয়, তাই ফাইলসিস্টেম সংক্রান্ত কমান্ডগুলো, ‘mkfs’ বা ‘fsck’ গোছের রাখা থাকে এই ডিরেক্টরিতে — এরা কী করে, মনে করুন তো। এই ‘mkfs’ বা ‘fsck’ প্রোগ্রামদুটো ‘mkfs.*’ বা ‘fsck.*’ চেহারাতেও থাকে, আলাদা আলাদা ধরণের তথ্যব্যবস্থা বা ফাইলব্যবস্থার জন্যে। ধরুন, ‘fsck.ext3’, ‘fsck.reiserfs’, ‘fsck.xfs’ বা ‘mkfs.ext2’, ‘mkfs.jfs’, ‘mkfs.xfs’। সিস্টেম বুট করার এবং বন্ধ বা শাটডাউন করার সময়কার প্রক্রিয়ায় কোথাও যাতে না আটকায়, প্রথম মাউন্ট এবং শেষ আনমাউন্ট তো করা হয় এই ডিরেক্টরিটাই, নানা ধরনের পার্টিশনে নানা কাজে সিস্টেম এই ডিরেক্টরি থেকে প্রোগ্রামগুলোকে কাজে লাগায়। স্ল্যাশের সিস্টেম বাইনারিদের মধ্যে, মানে ‘/sbin’ ডিরেক্টরিতে, গ্নু-

লিনাক্স ফাইলসিস্টেম হায়েরার্কি স্ট্যান্ডার্ড বা এফএসস্ট্যান্ড (FSSTND) অনুযায়ী কিছু বাইনারির অবশ্যই থাকা দরকার বা থাকার কথা, তাদের কয়েকটাকে আমরা এখানে তুলে দিলাম, দেখুন, এবং নিজেই এদের গুরুত্বটা বোঝার চেষ্টা করুন। রুট ডিরেক্টরির নিচের ডিরেক্টরিগুলোয় অন্য '/sbin' ডিরেক্টরির কথায় আমরা তারপর আসছি।

shutdown	সিস্টেমকে বন্ধ করে, নিয়ত চলমান প্রক্রিয়াগুলোকে পরপর ঠিকভাবে থামিয়ে
fdisk	পার্টিশন জুড়ে ফাইলসিস্টেম তৈরির আদেশ, আগেই বলেছি
fsck	পার্টিশনের ফাইলসিস্টেমে কোনো গড়বড় হয়েছে কিনা পরখ করে
getty	বুট করার সময় একজন ইউজারকে একটা টার্মিনালে ন্যস্ত করে
halt	সিস্টেমের কাজ বন্ধ করার আদেশ
ifconfig	নেটওয়ার্কের সঙ্গে সিস্টেমের পারস্পরিকতাটা যাচাই করে
init	বুট করার সময় মূল দায়িত্ব যে প্রোগ্রামের, পাঁচ নম্বর দিনে দেখুন
mkfs	একটা পার্টিশনে তথ্য রাখার জন্যে ফাইলব্যবস্থা তৈরি করে
mkswap	ডিস্কভূমির একটা অংশ সোয়াপের কাজে ব্যবহারের উপযুক্ত করে তোলে
reboot	রিবুট করে সিস্টেমকে, অফ করে অন করে
swapon	ডিস্কভূমির কোন অংশ সোয়াপের কাজে ব্যবহার হবে সেটা বলে দেয়
swapoff	ডিস্কভূমির কোনো একটা অংশকে সোয়াপের কাজে লাগানো বন্ধ করে
update	একটা যখ বা ডিমন, ফাইলসিস্টেম বাফারগুলোকে নিয়ম করে ডিস্কে লেখে

আমরা আগেই বিশদ ভাবে আলোচনা করেছি, একটা সিস্টেমে একটা ঐক্যবদ্ধ ফাইলসিস্টেমের এক একটা ডিরেক্টরি কী ভাবে এক একটা পার্টিশনে থাকতে পারে, যে পার্টিশনগুলো আলাদা আলাদা হার্ডডিস্কে এমনকি মেশিন গুলো নেটওয়ার্কাবদ্ধ থাকলে আলাদা আলাদা ভৌগোলিক অবস্থানেও থাকতে পারে। যদি '/usr' ডিরেক্টরি কোনো আলাদা পার্টিশনে থাকে, সেই পার্টিশন মাউন্ট হয়ে যাওয়ার পরে যেসব সিস্টেম কমান্ড কাজে লাগার কথা, তাদের রাখা থাকে '/usr/sbin' ডিরেক্টরিতে। বুট এবং শাটডাউনের গোটা কাজটাই যেমন হয়ে থাকে '/sbin' ডিরেক্টরি থেকে, সিস্টেমের কাজ করে চলাকালীন এই '/usr/sbin' ডিরেক্টরির গুরুত্ব বরং '/sbin' ডিরেক্টরির চেয়ে বেশি বই কম নয়। আমার মেশিনে সুজে এবং স্ল্যাকওয়ার সিস্টেমে এই দুটো ডিরেক্টরির মোট ফাইলের সংখ্যা দিয়ে তুলনা করুন। সুজেতে '/sbin' ডিরেক্টরিতে বাইনারি বা তার লিংকের মোট সংখ্যা ২২৫, সুজে '/usr/sbin' ডিরেক্টরিতে বাইনারি বা লিংকের সংখ্যা ২৬৩, আর স্ল্যাকওয়ারে '/sbin' ডিরেক্টরিতে ২০৮, '/usr/sbin' ডিরেক্টরিতে ২৭৮।

এর সঙ্গে তুলনা করুন '/usr/local/sbin' ডিরেক্টরিকে। আমার মেশিনে সুজে এবং স্ল্যাকওয়ার দুটো সিস্টেমেই এই '/usr/local/sbin' ডিরেক্টরিদুটোয় বাইনারির সংখ্যা একটা করে গৌরবজনক ০। কেন? কেন এদের এই সর্বব্যাপী শূন্যতা? অথচ দুটোরই রুট ব্যবহারকারী মানে সর্বময় 'root'-এর জন্যে '$PATH' সবিস্তারে দেখাচ্ছে এই ডিরেক্টরিটার নাম। গোটা নেটওয়ার্ক জুড়ে গ্লোবাল রকমে নয়, শুধু স্থানীয় একজন বা একাধিক ব্যবহারকারীর অধিকারের এলাকার দেখভালের জন্যে লোকাল রকমে যেসব সিস্টেম কমান্ড ইনস্টল করা হয় তাদের থাকার কথা '/usr/local/bin' ডিরেক্টরিতে। আমরা আগেই বলেছি, নানা ধরনের সিস্টেমে নানা ব্যবস্থা থাকে, আলাদা আলাদা ডিরেক্টরি আলাদা আলাদা পার্টিশনে মাউন্ট করা যায়।

১.৩।। '/usr' ডিরেক্টরি

আট নম্বর দিনের ৬.১ নম্বর সেকশনের টেবিলটার থেকে দেখুন, আমার মেশিনের সুজে আর স্ল্যাকওয়ার দুটো সিস্টেমেই '/usr' ডিরেক্টরির গুরুত্বটা। সুজেতে ৩.৫ জিবির মধ্যে ২.৫ জিবি, মানে একাত্তর শতাংশ, আর স্ল্যাকওয়ারে ২.৩ জিবির মধ্যে ১.৯ জিবি মানে প্রায় তিরাশি শতাংশ। এতটা একাই-খাবো অবশ্য হওয়ার কথা না '/usr' ডিরেক্টরির, আসলে আমি আমার কাজের প্রায় গোটাটাই রাখি '/mnt/arkive' ডিরেক্টরিতে, যেটা একটা রাইজারএফএস পার্টিশন, যাকে সুজে আর স্ল্যাকওয়ার দুজনেই মাউন্ট করতে পারে। আর ডকুমেন্ট বইপত্র এসব রাখি উইন্ডোজ পার্টিশনদুটোয়, লিনাক্স পার্টিশনে থাকলে উইন্ডোজ থেকে খোলা তো দূরের কথা দেখাই যাবেনা, আগেই বলেছি। তবে কোনো বেয়াড়া ব্যবহারকারীর কল্যাণে '/home' ডিরেক্টরি বেধড়ক বেড়ে না-গেলে '/usr' ডিরেক্টরি সবসময়ই গোটা সিস্টেমের ডিস্ক-ব্যবহারের সিংহভাগ জুড়ে থাকার কথা। যেমন দেখুন, '`df -h`' কমান্ড দিয়ে আমার কোন

হার্ডডিস্কের কোন পার্টিশনে কতটা ডিস্কভূমি ব্যবহার হয়েছে তার তালিকা থেকে দেখলাম, '/mnt/arkive' ডিরেক্টরি যেখানে মাউন্ট সেই '/dev/hdb5' পার্টিশনে মোট ডিস্কভূমি ২৫.৮ জিবির ভিতর ব্যবহৃত হয়েছে ২১ জিবি, আর উইন্ডোজ ডিরেক্টরি '/mnt/windows/c' আর '/mnt/windows/d' দুটো যেখানে মাউন্ট সেই '/dev/hda1' আর '/dev/hdb5' পার্টিশন দুটো মিলিয়ে মোট জমি ১৮.৭ জিবির মধ্যে মোট ব্যবহার হয়েছে ১২.৪ জিবি। এবার এর মধ্যে উইন্ডোজ অপারেটিং সিস্টেমের ফাইল আছে কিছুটা। ধরুন উইন্ডোজ 'C:'-এর মধ্যে সরাসরি রুট ডিরেক্টরিতে মানে আমাদের '/mnt/windows/c' ডিরেক্টরির মধ্যে যে ফাইলগুলো আছে, সেগুলো, আর '/mnt/windows/c/windows' ডিরেক্টরি এবং '/mnt/windows/c/Program\ Files' ডিরেক্টরির মধ্যে মোট ফাইল আর সাবডিরেক্টরি মিলিয়ে ব্যবহৃত হয়েছে ৮৭৭ এমবি। একটা জিনিষ খেয়াল করুন আমরা উইন্ডোজ-এ এক্সপ্লোরার দিয়ে যে ডিরেক্টরিটাকে দেখি 'Program Files' নামে, সেটাকে এখানে দেখছি 'Program\ Files' নামে। মধ্যের স্পেসটার আগে একটা '\' চিহ্ন দিতে হয়েছে। নইলে ব্যাশ শেল স্পেসটার আগে ও পরে দুটো শব্দ বলে ভাবত — এগুলো নিয়ে আমরা আজই কথা বলব। এবার আর্কাইভ আর উইন্ডোজ মিলিয়ে আমার যে মোট ডিস্ক ব্যবহার, ৩৩.৪ জিবি, এর গোটাটাই কিন্তু, আদতে থাকার কথা আমার সুজে আর স্ল্যাকওয়ার সিস্টেমের হোম ডিরেক্টরিদুটোয়, 'dd', 'manu' আর 'piu' ব্যবহারকারীদের নিজের নিজের ঘরে। অতিথির ঘরটা, এমন ব্যবস্থা করা আছে, প্রতিবার শাটডাউনের আগে ফাঁকা করে দেওয়া হয়, পরেরবার নতুন অতিথি আসার আগে ঘর পরিষ্কার করার মতন। তাই, হোম ডিরেক্টরি '/home' মেলালে, '/usr' ডিরেক্টরি তুলনায় অনেক শুঁটকো দেখায়, কিন্তু, হোম ডিরেক্টরি তো সিস্টেমের বানানো নয়। সেই অর্থে, সিস্টেমে সবচেয়ে ঘেমো এবং জাঁদরেল এই '/usr' ডিরেক্টরিটাই। আমার নিজের মনে আছে প্রথমবার এখানে ঢুকে কিরকম পথ হারিয়ে ফেলার অনুভূতি হয়েছিল। এবার এই ডিরেক্টরিকে একটু আলগা করে চিনব আমরা, শুধু একটা জিনিষ খেয়াল করিয়ে নিই। এইমাত্র দেখলাম, একটা উইন্ডোজ অপারেটিং সিস্টেমে মোট সিস্টেম ফাইলের কমবেশি পরিমাণ ৮৭৭ এমবি দাঁড়াচ্ছে। আর সুজে এবং স্ল্যাকওয়ারের মোট সিস্টেম ফাইলের পরিমাণ খেয়াল করুন, হোম ডিরেক্টরি দুটো বাদ দিয়ে, আট নম্বর দিনের ৬.১ নম্বর সেকশনের তালিকাটা মেলান, সুজে আর স্ল্যাকওয়ার এই দুটো সিস্টেমে হোম ডিরেক্টরি দুটো বাদ দিলে মোট ডিস্কজমি ব্যবহৃত হয়েছে যথাক্রমে ৩.৩৯ জিবি আর ২.২৯ জিবি। উইন্ডোজ-এর সঙ্গে তফাতটা খেয়াল করুন, আকারের পার্থক্যের কারণ এই যে, গ্নু-লিনাক্স কমিউনিটি আপনার কম্পিউটার শেখার জানার বোঝার কাজ করার প্রতিটি প্রত্যেকটি প্রয়োজনীয় জরুরি অংশ আপনার ডিস্ট্রোর মধ্যেই দিয়ে দিচ্ছে। এবং তাদের সংখ্যা আকার বৈচিত্র সবই উইনডোজ পরিবেশের চেয়ে সব অর্থেই অনেক বড়। ওই '/mnt/arkive' ডিরেক্টরির, মানে আর্কাইভের একটা বড় জায়গা জুড়ে রয়েছে এভিআই মানে ভিসিডির চেয়ে অনেক উন্নত এবং আকারে ছোট একটা ফর্ম্যাটে মাল্টিমিডিয়া ফাইল, তাদের দেখা, বদলানো থেকে শুরু করে, গত এক দশকে উইন্ডোজ এবং তার অফিস প্যাকেজ দিয়ে যত কাজ করেছি, সেই কাজগুলোর উপর নতুন কাজ করে চলার, গান শোনার, স্লাইড বানানোর, অংক কষার, প্রোগ্রাম লেখার, কম্পাইল করার, হিশেব করার, খেলার, ডিকশনারির, প্রতিটি প্রতিটি সফটওয়ার আমাদেরই বানানো, আমরাই কাজ করছি, আমরাই বদলাচ্ছি। এটা আমাদের সম্পত্তি, শুধু সম্পত্তিবোধটা আলাদা, বেজায় আলাদা, হেগেলের ফিলোজফি অফ রাইটের মরালিটি যা দিয়ে আমরা এতদিন বুঝে আসছি, তার বাইরে। যাকগে, মুজতবা আলির রকমে বললে, ফার্সিতে, 'খয়ের', বাজে বকা বাদ দ্যান, এবার আসা যাক '/usr' ডিরেক্টরির কথায়।

'/usr' ডিরেক্টরির এমন পুরুষ্টু রূপের, গাবদা গতরের, গোপন রহস্য কী? আসলে সিস্টেমের যাবতীয় ইউজারের ব্যবহারযোগ্য বাইনারি, তাদের ডকুমেন্টেশন, তাদের লাইব্রেরিগুলো, তাদের হেডার ফাইলগুলো, সবই থাকে এই ডিরেক্টরিতে। গুই মানে এক্স-উইনডোজ চালানোর সমস্ত সহায়ক লাইব্রেরিগুলোও থাকে এখানে। নেটওয়ার্কিং সংক্রান্ত যে প্রোগ্রামগুলো ব্যবহারকারীদের লাগে, 'telnet' বা 'ftp' জাতীয়, সেগুলোও থাকে এই একই ভুঁড়ে পেটে। আমরা আগেই বলেছি, আমাদের বর্তমান আসলে তার শরীরে অতীত নিয়ে বসে থাকে, ইতিমধ্যেই সদাসর্বদাই, অলওয়েজ অলরেডি, আমরা খেয়াল করি আর না-করি। পুরোনো দিনের ইউনিক্স গঠনে এই ইউজার বা '/usr' ডিরেক্টরির ভিতরেই থাকত প্রতিটি ব্যবহারকারীর নিজস্ব ঘর, মানে, আমার সিস্টেমের '/home/dd', '/home/manu' ইত্যাদি ডিরেক্টরিগুলোর মালমশলা একসময় থাকত এই ইউজারেই, তখন ঠিকানা থাকত '/usr/dd', '/usr/manu'। এখন এদের একত্রে আলাদা করে একটা কোয়ার্টার দেওয়া হয়েছে — '/home' বা হোম ডিরেক্টরি। এখন ইউজারে

শুধু ইউজারদেশের প্রোগ্রাম আর তথ্যই থাকে। মানে ইউজারের অবস্থানটা বদলে গেছে, ইউজার-সংক্রান্ত সবকিছু থেকে ইউজার-ব্যবহারযোগ্য-প্রোগ্রাম-সংক্রান্ত সবকিছুতে। এর মধ্যেকার সাবডিরেক্টরিগুলোকে একটু চিনে নেওয়া যাক।

/usr/X11R6 — এই সুবৃহৎ সাবডিরেক্টরিতে থাকে এক্স-উইনডোজ ব্যবস্থার জন্যে প্রয়োজনীয় সমস্ত লাইব্রেরি, সমস্ত চালনীয় বাইনারি, সমস্ত ডকুমেন্ট, ফন্ট এবং আরো বহু কিছু। কিন্তু এক্স-ব্যবস্থায় যা যা প্রোগ্রাম চলে তাদের সকলের সবকিছুই কিন্তু এখানে থাকেনা। যেমন, খুব জনপ্রিয় দুটো এক্স-উইনডোজ ব্যবস্থা হল কেডিই (KDE) আর গুহনোম (GNOME)। উইনডোজ থেকে আসা নতুন লিনাক্সীদের এটায় একটু অস্বস্তি হয় — উইনডোজ আবার নানা রকম মানে কী? গ্নু-লিনাক্সে বহু ধরনের উইনডোজ হয়, সবচেয়ে জনপ্রিয়দের মধ্যে এই দুটো ছাড়াও আছে ফ্লাক্সবক্স, ব্ল্যাকবক্স, এনলাইটেনমেন্ট ইত্যাদি। এই প্রত্যেক ধরনের এক্স-উইনডোজ ব্যবস্থায় উইনডোজের আকার কী হবে, কোথায় চলতে থাকা খোলা থাকা উইনডোগুলোর আইকনদের রাখা হবে, কোথা থেকে তাদের চালানো যাবে, চেহারা কী হবে, কী করে সবকিছু বদলানো যাবে, চলতে থাকা একটা উইনডো থেকে আর একটা উইনডোয় যাব কী করে — এই সমস্ত কিছুই আলাদা। এর মধ্যে কোনটা আপনি ব্যবহার করবেন সেটা আপনার পছন্দ। সায়মিন্দু সঙ্কর্ষণ যেমন মূলত গুহনোমপন্থী, অরিজিত তথাগত কেডিইপন্থী, আমি গুই ভালো বুঝি-না, বা পারি-না, যতটুকু করি তাতে মূলত ফ্লাক্সবক্সপন্থী, এইরকম। এই কেডিই গুহনোম ইত্যাদিদের কেজো ফাইলগুলো কিন্তু আবার এই '/usr/X11R6' সাবডিরেক্টরিতে থাকেনা। তারা সরাসরি '/usr' বা অন্য কোথাও রাখে। এক্স-উইনডোজের ডকুমেন্টেশন '/usr/X11R6/doc' ডিরেক্টরিতে থাকেনা, থাকে '/usr/X11R6/lib/X11/doc' ডিরেক্টরিতে। এইরকম কিছু এলোমেলোপনা আছে। এই কিছুটা এলোমেলো থাকার কারণ কিন্তু গ্নু-লিনাক্সের গতিশীলতা, সেটা প্রতি মুহূর্তেই বাড়ছে বদলাচ্ছে নড়ছে, থিতু হওয়া আর ফের পায়ের তলার সর্ষে কুটকুট করা — দুটোই চলছে একই সঙ্গে। এই '/usr/X11R6' ডিরেক্টরির মধ্যে আবার কিছু সাবডিরেক্টরি আছে। যেমন '/usr/X11R6/bin' ডিরেক্টরি — এখানে থাকে এক্স-উইনডোজ সংক্রান্ত বাইনারিগুলো, যে প্রোগ্রামগুলো এক্স-উইনডোজ চালু করে, কনফিগার করে, এবং চালায়। 'X', 'xf86config', 'xauth' জাতীয়। এই প্রোগ্রামগুলো ঠিক কিসের, রবীন্দ্র-নজরুল না জীবনমুখী না নির্বাচনী — সেসব তো, হুঁহুঁ বাবা, আগেই বলে নিয়েছি, গুই এই পাঠমালার কাজ না, তারপরে নিজেই মরি আর কী, কোনোক্রমে আন্দাজে উইনডোজের কাজ করি, কিন্তু কোনোক্রমে আন্দাজে তো লেখা যায়না। '/usr/X11R6/include' ডিরেক্টরিতে থাকে হেডার ফাইলগুলো, হেডার ফাইল কাকে বলে মনে আছে? এক নম্বর দিনে বলেছিলাম, তারপরেও এসেছে প্রসঙ্গটা। এগুলো লাগে প্রোগ্রামিং ভাষায় লেখা কোনো কোডকে কম্পাইল করে প্রোগ্রাম বানানোর সময়। এক্স টুলকিট বলা হয় এক্স-উইনডোজের নানা যন্ত্রপাতিকে, সেগুলোকে কাজে লাগায় এমন প্রোগ্রাম কম্পাইল করতে গেলে প্রয়োজন পড়ে এই '/usr/X11R6/include' ডিরেক্টরির হেডারদের। সিস্টেম লাইব্রেরিও, মনে করুন, এক ধরনের বারবার ব্যবহারযোগ্য ফাংশনের ভাণ্ডার, এদেরও লাগে ওই প্রোগ্রামিং এর কাজেই, এক্স উইনডোজ প্রোগ্রামিং-এর লাইব্রেরিগুলো থাকে '/usr/X11R6/lib' ডিরেক্টরিতে। ফন্ট মানে একই বর্ণমালার নানা চেহারার নানা অবতার-পরিবার। ধরুন 'GNU-Linux' বা '`GNU-Linux`' বা 'GNU-Linux'। এদের তিনটেতেই আছে একই বর্ণমালা, কিন্তু আলাদা আলাদা তিনটে ফন্টে — টাইমস, কুরিয়ের আর হেলভেটিকা। প্রত্যেকটা ফন্টের পরিবারের মধ্যে বর্ণমালাগুলোর মধ্যে একটা মিল এবং আত্মীয়তা থাকে। যেমন ধরুন টাইমস আর কুরিয়েরের সেরিফ আছে, মানে, 'i' বা '`i`' অক্ষরটার উপর নিচে দুটো আনুভূমিক দাগ, কিন্তু হেলভেটিকার 'i'-তে নেই। আবার টাইমস এবং হেলভেটিকার সঙ্গে কুরিয়েরের পার্থক্য এই যে, কুরিয়েরে প্রতিটি বর্ণ একই পরিমাণ অনুভূমিক জমি নিয়ে রয়েছে, সে তন্দুরস্ত '`G`' হোক বা ডায়েটিনী '`i`', এইজন্যে একে বলে কন্সটান্ট-উইডথ বা সমঔদার্য ফন্ট। যান্ত্রিক টাইপরাইটারগুলোর যেমন। এই বিষয়ে বিপুল ফান্ডা লড়িয়ে প্রেসের লোকদেরও নার্ভাস করে দিতে পারবেন, হাউটু-মালা থেকে ফন্ট হাউটুগুলো পড়ে নিলেই। এক্স উইনডোজে ব্যবহারের সিস্টেম ফন্টগুলো রাখা থাকে '/usr/X11R6/lib/X11/fonts' ডিরেক্টরিতে। এই ফন্টগুলোকে চালু করে এক্স স্ক্রিন ফন্ট বা 'xfs' নামে একটা যখ, সিস্টেম চালু হওয়ার সময় দেখবেন, স্ক্রিনে ফুটে ওঠে। এর সঙ্গে কিন্তু ফাইলসিস্টেমের এক্সএফএস গুলিয়ে

ফেলবেন না। আমি নিজে একবার ফেলেছিলাম, একদম প্রথম দিকে, আমার এক্সএফএস ফাইলসিস্টেম তখন নেই, তাহলে ব্যাটা এক্সএফএস ডিমন চালু করছে কেন?

/usr/bin — এই ডিরেক্টরিতে কী থাকে, সেটা আপনি আন্দাজ করতে পারছেন। ইউজার বাইনারি। অত্যাবশ্যকীয় ইউজার বাইনারিগুলো থাকে '/bin' ডিরেক্টরিতে, আমরা আগেই বলেছি। তার বাইরে আপনি আর যা যা বাইনারি মানে প্রোগ্রাম ব্যবহার করবেন তার প্রায় সবগুলোই থাকে এই '/usr/bin' ডিরেক্টরিতে। হয় তারা নিজে নিজে সগৌরবে, নিদেন তাদের পাদুকা, মানে লিংক। নিজে ঢুকে একবার দেখুন, প্রথমদিকে দেখবেন, দু-একটা নাম কেবল চিনতে পারছেন, 'emacs', 'fortune', 'gcc' ইত্যাদি। ধীরে ধীরে চিনে যাবেন। এই 'ধীরে'-টা কিন্তু জেনুইন ধীরে, কারণ, এই ডিরেক্টরিতে মোট কত বাইনারি থাকে বলুন তো? একটা আন্দাজ দিই। আমার মেশিনের সুজে সিস্টেমে '/bin' ডিরেক্টরির মোট বাইনারি বা তাদের লিংকের সংখ্যা ৮৪, আর '/usr/bin' ডিরেক্টরিতে ওই সংখ্যাটা ১৭৯৫। যদি ভাবেন এটা সুজের ইউজার-মারার কল, তা নয়, স্ল্যাকওয়ারে ওই সংখ্যাদুটো ৯৬ আর ১৭৩২। তাও আমি তো গ্নু-লিনাক্স সিস্টেমে একদমই হাফটিকিট ইউজার, আমার মেশিনেই এই, তেমন কম্পিউটার জানা লোকজনের মেশিনে কী হবে ভাবতে পারছেন?

/usr/doc — নাম থেকেই বুঝতে পারছেন, ডকুমেন্টেশন। সেটার মধ্যে প্রথামাফিক থাকার কথা লিংক, মূল ডকুমেন্টেশনটা থাকার কথা '/usr/share/doc' ডিরেক্টরিতে। এই লিংক ফিংক রাখাটা যে মানতেই হবে তা নয়, কেউ মানে, কেউ মানে না। স্ল্যাকওয়ারে হুবহু একই ডকুমেন্টেশন রাখা আছে '/usr/doc' আর '/usr/share/doc' ডিরেক্টরি দুটোতে। দুটোরই সাইজ ১৪২ এমবি। এটা আমি আগে কোনোদিন খেয়াল করিনি। আজ আপনাদের এটা লেখার আগেই দেখলাম এই অদ্ভুত ব্যাপারটা। আর সামান্য কিছু ডকুমেন্টেশন রাখা '/usr/local/share/doc' ডিরেক্টরিতে, ১৬৮ কেবি। এটা স্ল্যাকওয়ার ভারশন ৮.২। তথাগত স্ল্যাকের ভারশন নাইনের আইএসও ইমেজদুটো নামিয়ে দিয়েছে, সেটা ইনস্টল করে দেখতে হবে সেখানেও এই একই কিনা। এই বইটা শেষ হওয়ার আগে হাত দিতে পারছিনা। সুজেতে '/usr/doc' আর '/usr/local/share/doc' দুটো ডিরেক্টরিই বেশ বিনয়ী। যথাক্রমে ৫৬ কেবি আর ১.৩ এমবি। বেদনাটা আছে '/usr/share/doc' ডিরেক্টরিতে। ৪০৬ এমবি। এইটা শুনে আন্দাজ করা খুব কঠিন, ঠিক কতটা ডকুমেন্টেশন। একটা আন্দাজ দেওয়ার চেষ্টা করা যাক। জিএলটির অশোকদা আমাকে প্রোজেক্ট গুটেনবার্গের গোটা সিডিটা ডাউনলোড করে পুড়িয়ে দিয়েছে, তার থেকে জেন আয়ার উপন্যাসটা ধরুন, একহাজার কেবি সাইজ। এটাকে স্বাভাবিক ১২ পয়েন্ট টাইমস ফন্টে এফোর সাইজের পাতায়, মানে সচরাচর আমরা যে সাইজের পাতায় পড়ি তার মোটামুটি দ্বিগুণ সাইজে, ২১০ বাই ২৯৭ মিলিমিটারে, আসছে ৩৭৩ পাতা। এটায় খুব বেশি ফরম্যাটিং নেই, ফরম্যাটিং থাকলে সাইজ বাড়ে, লিনাক্স ডকুমেন্টেশনে ফরম্যাটিং এর চেয়েও কম। এই হারে হিশেব করলে মোটামুটি ভাবে আসছে সুজের '/usr/share/doc' ডিরেক্টরির ৪০৬ এমবি মানে এফোর দেড় লাখ পাতার মত, মানে স্বাভাবিক পাতায় তিন লাখের কাছাকাছি। এবং এখানেই বিস্ময়টা শেষ নয়, এর গোটাটাই আমাদের করা — আ-মা-দে-র — গ্নু-লিনাক্স কমিউনিটির দ্বারা এবং জন্যে। পয়সা আয়ের জন্যে নয়, নিজের জানাটা অন্যান্য নিজের লোকদের সঙ্গে বাঁটোয়ারা করার জন্যে। ভালোবাসা এবং গ্নু-লিনাক্স — পৃথিবীর শেষ দুটো ম্যাজিক এখানে একসঙ্গে কাজ করছে। এই ডিরেক্টরির মধ্যে কতটা করে এবং কী ডকুমেন্টেশন আছে '`du -chs /usr/share/doc/*`' মেরে দেখে নেওয়া যাক।

৪৫২ কেবি	/usr/share/doc/NVIDIA_GLX-1.0	১০০ কেবি	/usr/share/doc/release-notes
৯২ কেবি	/usr/share/doc/bladeenc-0.94.2	৩০ এমবি	/usr/share/doc/sdb
৫.৪ এমবি	/usr/share/doc/glibc	১.৪ এমবি	/usr/share/doc/susefaq
৮৭ এমবি	/usr/share/doc/howto	৪.৫ এমবি	/usr/share/doc/susetour
৪.৬ এমবি	/usr/share/doc/kernel	১৫২ কেবি	/usr/share/doc/vcdimager-0.6.2
২৭৪ এমবি	/usr/share/doc/packages	৪০৬ এমবি	মোট

এর মধ্যে দেখুন সবচেয়ে আসুরিক '/usr/share/doc/packages', এটাতে ভরা থাকে সমস্ত প্যাকেজের আলাদা আলাদা ডকুমেন্টেশনগুলো, প্যাকেজের নামে নামে আলাদা আলাদা ডিরেক্টরিতে। দুনম্বর সাইজের ডার্লিংটিকে তো আপনি আগেই চেনেন, '/usr/share/doc/howto', বেশ কয়েকদিন হল ডেট করে আসছেন। এর মধ্যে সুজের

নিজস্ব খ্যাঁচ আছে তিনখানা, ‘sdb’, ‘susefaq’ এবং ‘susetour’। এর মধ্যে তিন নম্বরটা হল পর্যটন, ইনস্টল করার পরে আপনাকে ঘুরিয়ে ঘুরিয়ে দেখাবে, বহুদিন আগে আমি একসময় উইন্ডোজ ৯৫ এবং এমএস ওয়ার্ড লাইসেন্সড প্যাক কিনেছিলাম, তার সঙ্গে একটা যুধিষ্ঠিরের নরকদর্শন গোছের সিডি ভালবেসে আমায় উপহার দিয়েছিলেন স্যামকাকুর ভাই গেটসকাকু, ‘হোয়ার ডু ইউ ওয়ান্ট টু গো টুডে’। পাঁঠবিকতায় প্রায় তার কাছাকাছি এই সুজেটুর, দেখতে দেখতে মনে হয়, যাঃ শালা, এত কষ্ট করে গোপন করার পরেও এরা জানল কী করে আমার আইকিউ আশির নিচে। আর ওই ‘NVIDIA_GLX-1.0’ হল আমার মেশিনের ভিডিও ড্রাইভারের নিজস্ব। এই সমস্ত ডকুমেন্টেশন ছাড়া আর যা থাকে সিস্টেমে, ‘info’ আর ‘man’, সেগুলো আগে থাকত ‘/usr/info’ আর ‘/usr/man’ ডিরেক্টরিতে, এখন থাকে ‘/usr/share/info’ আর ‘/usr/share/man’ ডিরেক্টরিতে। এর প্রত্যেকটা ডিরেক্টরিতেই একবার ঢুকুন আর দেখুন। মনে থাকবেনা গোটাটা, কিন্তু, পরিচিতিটা বাড়বে। পরে, সিস্টেমের মালিক এবং দাস হয়ে ওঠার প্রক্রিয়ায়, দুটোই একসঙ্গে হতে হয়, আলাদা করে একটা হওয়া যায়না, মাথায় সিস্টেমের শালগ্রাম শিলা নিরন্তর বহনের সুবিধে হবে। গোটাটাই করে চলুন ‘`cd`’ আর ‘`ls`’ কমান্ড দুটো ব্যবহার করে।

/usr/etc — কনফিগারেশন ফাইলগুলো আগে একসময় এখানে রাখা হত, এখন আর ব্যবহার হয়না। অনেকটা এরকমই আর একটা ডিরেক্টরি ‘/usr/games’, সেটা তবু এখনো কচিৎ কদাচিৎ ব্যবহার হয়।

/usr/include — এটা আন্দাজ করে নিতে পারছেন, এখানে ইউজার জগতের সোর্সকোড কম্পাইল করার জন্যে দরকারি হেডার ফাইলগুলো থাকে। এর মধ্যে আবার আলাদা আলাদা প্যাকেজের নামে আলাদা আলাদা ডিরেক্টরিত থাকতে পারে, যেখানে সেই প্যাকেজের জন্যে দরকারি হেডার ফাইলগুলো। যেমন আপনার একটা সাপ্লিমেন্টারি মুন্ডু কম্পাইল করে নেওয়ার জন্যে যে হেডার ফাইলগুলো লাগবে, যদি ‘mundu.tar.bz2’ বা ‘mundu.tar.gz’ ইত্যাদি থেকে মানে ক্রাঞ্চ করে রাখা সোর্সকোড থেকে ইনস্টল করেন, সেগুলো থাকবে ‘/usr/include/mundu’ ডিরেক্টরিতে। আর যদি প্রিকম্পাইলড বাইনারি মুন্ডুতে, আরপিএম গোছের, আপনার কাজ চলে যায়, তাহলে তো ল্যাটা চুকেই গেল। ‘/usr/lib’ ডিরেক্টরিতে থাকে প্রোগ্রাম চালানোর হরহামেশা ব্যবহৃত লাইব্রেরিগুলো।

/usr/local — এই ডিরেক্টরির কথা আগেও এসেছে, বাইনারি ফাইলগুলো কোথায় থাকে সেই আলোচনায়। নিজে নিজে কম্পাইল করে নেওয়া বহিরাগত প্রোগ্রামগুলো রাখার একটা ভালো জায়গা। এইরকম কোনো প্রোগ্রাম ইনস্টল করার সময়ে সিস্টেমনিয়ন্তা বা রুট নিশ্চিন্তমনে তাদের এখানে রাখতে পারে, এতে একটা ভার্শন থেকে আর একটা ভার্শনে আপগ্রেড করার সময়ে বা অন্য কোনো আপডেটের সময়ে এদের বদলে যাওয়ার কোনো চান্স থাকেনা।

/usr/sbin — এই ডিরেক্টরিতে থাকে বাইরে থেকে যোগ করা বিভিন্ন সিস্টেমনিয়ন্ত্রক বা সিস্টেম অ্যাডমিনস্টারিং প্রোগ্রামের রুটযোগ্য সিস্টেমবাইনারিগুলো, ঠিক ‘/sbin’ ডিরেক্টরির মত এটাও শুধু রুটের ‘$PATH’-এ থাকে, অন্য ইউজারের পথনির্দেশে থাকেনা।

/usr/share — এই ডিরেক্টরির মধ্যে আবার অনেকগুলো সাবডিরেক্টরি আছে, ‘/usr/share/doc’, ‘/usr/share/info’, ‘/usr/share/man’ ইত্যাদি। ছয় নম্বর দিনে আমরা ম্যানপেজগুলোর নম্বর নিয়ে যে কথাগুলো বলেছিলাম, সেগুলোকে এবার মিলিয়ে নিন এই ‘/usr/share/man’ ডিরেক্টরির মধ্যে ঢুকে। ‘/usr/share’ ডিরেক্টরির মধ্যের কাঠামোর কোনো কোনোটা নিয়ে একটু একটু কথা হয়েছে, অবশিষ্ট জায়গাটা আপনার জন্যে রইল, আগের গুলোর সঙ্গে মিলিয়ে মিলিয়ে নিজেই ব্রাউজ করে দেখুন। আপনার নিজের পয়সায় কেনা হার্ডডিস্ক, তার ব্লকে ব্লকে কী মাল ছড়ানো আছে সেটা একবার মিলিয়ে নিতে হবেনা?

/usr/src — এই ডিরেক্টরিটা আমাদের শেষ মনোযোগের জায়গা, এর পরেই আসবে ‘/etc’ ডিরেক্টরির কনফিগারেশন ফাইলগুলোর কথা, তারপরেই আমাদের গ্নু-লিনাক্স ফাইলসিস্টেম হায়েরার্কির ব্যাপারটা শেষ, সবকিছুই তাহলে শেষ হয়। গ্নু-লিনাক্স সিস্টেমের এই ‘/usr/src’ ডিরেক্টরিটাতেই থাকে লিনাক্স কারনেল সোর্সকোড, তার হেডার ফাইলগুলো এবং ডকুমেন্টেশন। ধরুন বলাই যায়, এটাই সেই কৌটো গ্নু-লিনাক্সকে যা গ্নু-লিনাক্স করেছে। একটু আগে আমরা আরপিএম প্যাকেজের কথা বলেছি, যেগুলো প্রিকম্পাইলড বাইনারি, যাদের সোর্সকোড থেকে

বাইনারি ফাইল বা এক্সিকিউটেবল ফাইল ইতিমধ্যেই বানানো রয়েছে। আরপিএমের মধ্যে ব্যবস্থাটা এমন করা থাকে যে শুধু বাইনারি ফাইলটা নয়, তার সঙ্গে প্রয়োজনীয় লাইব্রেরি ফাইল ডকুমেন্টেশন ফাইল ইত্যাদি সমস্তই সঠিক সঠিক জায়গায় রেখে দেওয়া হয় আরপিএম প্যাকেজটা ইনস্টল করার সময়ে। এই আরপিএমকে সোর্স আকারেও পাওয়া যায়, এদের বলে সোর্স-আরপিএম। এই সোর্স-আরপিএম ইনস্টল করা মানে, যে প্রোগ্রামের সোর্স-আরপিএম ইনস্টল করছি, তার আরপিএমটা তৈরি করে নেওয়া, এই মেশিনের সমস্ত খুঁটিনাটির সঙ্গে মিলিয়ে, ঠিক কম্পাইল করার মত, এবং তার সঙ্গে আরপিএম সুবিধাটাও থাকছে, ও নিজেই সঠিক ফাইল সঠিক জায়গায় রেখে দেবে, আমায় কিছু ভাবতে হবেনা। সোর্স-আরপিএম থেকে আমরা যখন আরপিএম বানাই তখন সেই সংক্রান্ত ফাইলগুলো রাখা হয় '/usr/src' ডিরেক্টরির '/usr/src/RPM' সাবডিরেক্টরিতে। এর মধ্যে আবার অনেকগুলো সাবডিরেক্টরি আছে, যাদের আবার আলাদা আলাদা নিয়ম আছে। আর এই '/usr/src' ডিরেক্টরির মধ্যেই '/usr/src/linux' সাবডিরেক্টরিতে থাকে কারনেলের সোর্স কোড। তার মধ্যেকার অনেকগুলো সাবডিরেক্টরির ভিতর একটার কথা আমরা আগেই উল্লেখ করেছিলাম, ফাইলসিস্টেম সংক্রান্ত তথ্য পাওয়ার জন্যে, সেটা হল '/usr/src/linux/Documentation'।

১.৪।। '/etc' ডিরেক্টরি

মা তারা, আর দেরি নেই, আমাদের ফাইলসিস্টেম হায়েরার্কি হয়ে এল, প্রায় হয়ে এল এই বইটাও। এই '/etc' ডিরেক্টরি হল সিস্টেমের স্নায়ুকেন্দ্র। সিস্টেম সংক্রান্ত যে কোনো কনফিগারেশন ফাইল, 'যে-কোনো' মানে একদম ইংরিজির 'এনি অ্যান্ড এভরি' বলতে যা বোঝায়, থাকে এই ডিরেক্টরিতে বা এর পেটে কোনো সাবডিরেক্টরিতে। এই কনফিগারেশন ফাইলগুলোর কিছু থাকে সরাসরি এই ডিরেক্টরিতেই, যার কয়েকটার সঙ্গে আমাদের আগেই মোলাকাত হয়েছে, যেমন '/etc/fstab' বা '/etc/password' ইত্যাদি। আরো কিছু ফাইল এবং ডিরেক্টরিকে এবার চিনব আমরা। এটা কিন্তু কখনোই পূর্ণাঙ্গ হবেনা, পুরোটা জানতে হলে আপনাকে বিন এনগুয়েনের ওই লিনাক্স ফাইলসিস্টেম হায়েরার্কি পড়তে হবে। গোটাটার একটু আন্দাজ দিয়ে রাখি, আমার মেশিনের সুজে সিস্টেমে গোটা '/etc' ডিরেক্টরিটার সাইজ ৪৮ মেগাবাইট, এবং তার মধ্যে ডিরেক্টরি আছে ৫৪-টা, লিংক আছে চারটে, এবং সাধারণ ফাইল আছে ১৬৪-টা। এই চুয়ান্নটা ডিরেক্টরি নয়, এর মধ্যে কয়েকটা মূল ডিরেক্টরি, গ্নু-লিনাক্স ফাইলসিস্টেম হায়েরার্কি স্ট্যান্ডার্ড মেনে যারা থাকতে পারে, তার দু-একটাকে দিচ্ছি এখানে। কনফিগারেশন ফাইলগুলোকে একসঙ্গে ধরছি পরের সেকশনে।

/etc/X11/ — একে আমরা আগে থেকেই চিনি। এক্স উইনডোজের হাল-হকিকত, তার নাড়িভুড়ি, তার পোস্টার ব্যানার, বন্ধুতাপূর্ণ রাজনৈতিক চিন্তাবিনিময়ের পাইপগান, গণতান্ত্রিক আন্দোলনের নখ থেকে কালি তোলার ইরেজিং ইংক, মায় মজদুরচেতনার ভ্যানগার্ডের গোপন প্রাইভেট সেলফোন — মানে এক কথায় তার অর্থ এবং পরমার্থের সমস্ত ছক থাকে এখানে। এই সব ছক মেনে ক্রমছড়ায়মান মানবিক শুভবোধ এবং গনতান্ত্রিক চিন্তাপ্রক্রিয়ার এবং তার রাজনৈতিক পথনির্দেশের মতই এক্স-উইনডোজও সর্বব্যাপী, সর্বত্রগামী। তাকে আপনি জেনে বুঝে শিখে যাবেন এমনিতেই, এমনকি না-চাইলেও। শুধু শুধু আমাদের এই টেক্সটে আর ওই বাহুল্য কেন? শুধু একটাই কথা, এই ডিরেক্টরির বহু ফাইলই আসলে আসল ফাইল নয়, মুল মাথার কান মাত্র, অবশ্য টানলেই আসলটা আসবে, মাথারা আছে '/usr/X11R6' ডিরেক্টরিতে। এই ডিরেক্টরির আবার প্রচুর গলির গলি তস্য গলি আছে, ঘুরে দেখুন, নানা ধরনের ইলেকট্রনিক তেলেভাজা মিলবে মোড়ে মোড়ে। এক্স-উইনডোজের আলোচনা এই বইটা থেকে বাদ দেওয়ায় একটা জায়গা নিয়েই আমার একটু দুঃখ হয়েছে — '/etc/X11/XF86Config' ফাইলের গঠন — কী যে আকর্ষণীয় তার খুঁটিনাটিগুলো সে আর বলার নয়। যাকগে, বেঁচে থাকা মানেই তো অপরিপূরিত সাধের ব্যাকআপ বাড়ানো, আলাদা করে আর সিডিতে পোড়াতে হয়না, এমনিতেই পুড়ে শরীরে বসে যায়।

/etc/cron.d — শুধু এই ডিরেক্টরিটা নয়, এর সঙ্গে থাকে '/etc/cron.daily', '/etc/cron.hourly', '/etc/cron.monthly' ইত্যাদি ডিরেক্টরি। এর সঙ্গে থাকে '/etc/crontab' বলে একটা কনফিগারেশন ফাইল। এরা সবাই হল 'cron' নামের একটা যখের কাজের হাল হকিকত। ক্রন ডিমন, এর নাম থেকে আন্দাজ করার চেষ্টা করুন, সঙ্গে ডিরেক্টরির নামগুলো মিলিয়ে, এক একটা বিশেষ সময়ে, বা একটা বিশেষ সময় অন্তর, কী কী কাজ

সিস্টেমের জন্যে করে রাখতে হবে সেইটা মাথায় রাখে। ডিমন বা যখদের কাজই তো তাই, আপনার খেয়ালের বাইরে সিস্টেমের আভ্যন্তরীন প্রসেস বা পদ্ধতিগুলোকে চালিয়ে নিয়ে চলা। এই প্রসেসদের নিয়ে আমরা বিশেষ ভাবে কথা বলেছি দুই নম্বর দিনে, এবং পরে, পাঁচ আর ছয়ে। ক্রনের একটা দোসর আছে, তার নাম অ্যানাক্রন। এদের সঙ্গে ভালো করে দোস্তি করে ফেলুন, যখের ধন, আবার যখের ধনের বিমল আনন্দ পেতে থাকবেন। সিস্টেম ডকুমেন্টেশনে দেখুন, সব আছে। দেখবেন, আপনার কাজের সঙ্গে মিলিয়ে, মানে সিপিইউ-র উপর আপনি কখন চাপ কম দিচ্ছেন সেটা মাথায় রেখে, আপনার দরকারি কাজগুলো করে রাখবে।

যেমন ধরুন ফাইল খোঁজার জন্যে 'locate' কমান্ডটার কথা বলেছি আমরা। এই কমান্ডটা কাজ করে একটা ডেটাবেস দিয়ে। এই ডেটাবেসে ভরা থাকে যাবতীয় ফাইলের যাবতীয় মেটাডেটা। এর সঙ্গে তুলনা করুন ফাইল খোঁজার আর একটা কমান্ড 'find'। ফাইন্ড-এর সঙ্গে লোকেট-এর তফাতটা এই যে, আপনি যখন কোনো ফাইল খুঁজতে চাইলেন, সে তার নাম দিয়ে হোক, বানানোর বা ব্যবহারের তারিখ দিয়ে হোক, বা, ফাইলের মধ্যে কী আছে তার ভিত্তিতে হোক, যখনি আপনি কমান্ডটা দেবেন, ফাইন্ড খুঁজতে শুরু করবে বাস্তব ঐক্যবদ্ধ ফাইলব্যবস্থার বাস্তব ডিরেক্টরিটা। আর লোকেট হল ইন্টেলিজেনশিয়া, সে মেটাডেটা পড়ে খুঁজে দেবে, তাই অবভিয়াসলি অনেক দ্রুত কাজ করে লোকেট। কিন্তু ওই ডেটাবেস। মনে পড়ছে, রুট হয়ে 'updatedb' কমান্ড দিয়ে ডেটাবেসটা আপডেট করে নেওয়ার কথা? এবার, আপনি তো মানুষ, যখ নন, প্রত্যেকবার ঠিক সময়ে আপনার খেয়াল নাই থাকতে পারে, এবার কখনো একটা ফাইল খুঁজতে গেলেন, দেখলেন ডেটাবেসটা বেশ পুরোনো, খুব সাম্প্রতিক বদলগুলো তাতে আসেইনি। বেশ তো, কাজটা যখের হাতে ছেড়ে দিন, সেই নিয়মিত প্রাত্যহিকতায় কাজটা করতে থাকবে। আর করবেও ঠিক আপনার কাজের জন্যে সিপিইউর উপর চাপটার সঙ্গে মিলিয়ে, সেই যে মাল্টিটাস্কিং মাল্টিপ্লেক্সিং-এর কথা বলেছিলাম, সেই ভাবে, আপনার কাজকে মূল প্রায়োরিটিতে রেখে, নিজের কাজটাকে অনেক পিছনে ঠেলে দিয়ে। আর আপনি 'crontab' নামের কনফিগারেশন ফাইলটা দিয়ে তাকে বিশেষ করে আদেশ দিয়ে দিতে পারেন, অমুক সময়টায় কাজটা করো, যে সময়টায় আপনার কোনোই কাজ নেই। তবে, এর মানে, সেই সময়টায় আপনার মেশিন অন থাকবে।

গ্নু-লিনাক্স যেমন ধরে নেয় আপনার মেশিন নেটওয়ার্কে আছে, এটাই ডিফল্ট, তেমনি আপনার মেশিন সবসময়েই অন আছে, এটাই ডিফল্ট। পাঁচ নম্বর দিনে পিটার সেলাসের যে বক্তৃতাটার থেকে অনেকগুলো তথ্য এনেছিলাম, সেইটায় একজায়গায়, উইনডোজ সিস্টেমের বারংবার রিস্টার্ট করতে হওয়ার কথা তুলে, সেলাস নিজের বিএসডি অপারেটিং সিস্টেমে তিন বছরের উপর চলতে থাকা মেশিনের কথা উল্লেখ করে বলেছিলেন, আমি তো বুঝিনা, সিস্টেম বন্ধ হবে কেন, যদিনা কোনো হার্ডওয়ার গন্ডগোল করে? এদের কাছে সিস্টেম অফ হতে পারে শুধু হার্ডওয়ার বসে গেলেই, এছাড়া সবসময়েই অন। প্রথম বিশ্বে এটা সম্ভব তার কারণই এই যে আমাদের কাছে এটা অসম্ভব। একটা কম্পিউটার খুব কমই বিদ্যুৎ পোড়ায়, মোটামুটি একশো থেকে দুশো ওয়াটের মধ্যে, বোধহয় — কিন্তু গোটা দিন রাত অষ্টপ্রহর সেটা চলা মানে বিদ্যুৎ খরচ যেটুকু বাড়া সেটা আমার অর্থনৈতিক অবস্থায় সম্ভব নয়। তাও, আমার পরিজনদের মধ্যে, সে জিএলটি হোক, পলিটিকাল ইকনমি হোক, আমার অবস্থা অনেকের চেয়েই ভালো। ঠিক ওই নেটের মতই, প্রথম বিশ্বে দিবারাত্রি ওরা নেট পায়, অনেক দ্রুতগতি ওদের সংযোগ।

আমার এক মার্কিন বন্ধু, ক্রিস, সে তখন একটু মানসিক মন্দাতেও ছিল, যখনি নেটে কানেক্ট করছি দেখছি ও-ও অনলাইন, অথচ কোনো ইনস্ট্যান্ট মেসেজ বা মেলের উত্তর দিচ্ছেনা, খুবই চিন্তা হচ্ছিল। পরে জানলাম ও কয়েকদিনের জন্যে কোথাও চলে গেছিল, ভুলবশত নেট কানেক্ট করে রেখে। এই ভুলটা যদি আমি করতেও চাইতাম, পারতাম না। আমরা তৃতীয় বিশ্ব। নগন্য পরিমাণ নেট করেও আমার মাসিক টেলিফোন বিল আসে দুহাজার টাকার মত, যা, লোন কাটার পরে হাতে পাওয়া মাইনের প্রায় একের পাঁচ ভাগ। ঠিক এই সীমাবদ্ধতাগুলোই আমাদের আরো বেশি করে ভাবিয়েছিল জিএলটির কথা, যে যতটুকু যোগাড় করতে পারছি — জানা, তথ্য, ডাউনলোড, চিন্তা — সবকিছু একজায়গায় আসুক, একটা সাধারণ ভাণ্ডার তৈরি হোক। কিছুটা তো তবু বেশি পাব প্রত্যেকেই। এই লেখাটাও তো সেখান থেকেই, আমি মধ্যগ্রাম জিএলটির রিসোর্স পার্সন। জিএলটির সিডিটায় যে রিসোর্সগুলো রেখেছি, একটা ওয়েবসাইটের আকারে, সেগুলো অনেকটা ডাউনলোড করা জিএলটির হয়ে অশোকদার আর আমার, কিছুটা লাগের অরিজিত তথাগত সঙ্কর্ষণের। ডিস্ট্রোর সিডিগুলোও তাই। কেউ এখান থেকে দিয়েছে, কেউ ওখান

থেকে। তাও, বেশিরভাগ ছেলেপুলের যা অবস্থা, সিডি কিনতে বলতেও খারাপ লাগে। নিজেই, হাতখরচ বাঁচিয়ে, কিছু কিছু করে সিডি কিনে রাখতে হয়। যাকগে, '/etc' ডিরেক্টরির কথায় ফেরত আসা যাক।

/etc/cups — এই ডিরেক্টরির মধ্যে থাকে কাপস (CUPS — **C**ommon-**U**nix-**P**rinting-**S**ystem) বা প্রিন্ট সংক্রান্ত যাবতীয় কনফিগারেশন ফাইল। এটা খুব গুরুত্বপূর্ণ একটা জায়গা, ভালো করে ডকুমেন্টেশন পড়ে নিন। সঙ্গে গোস্টস্ক্রিপ্ট বা জিএস-এর (gs) ম্যানুয়ালও পড়ুন। আমায় গোস্টস্ক্রিপ্ট পড়ার কথা বলেছিলেন লাগ-এর মানস লাহা। সত্যিই, ম্যান জিএস করে দেখুন, প্রিন্ট সংক্রান্ত বোধহয় এমন কিছু নেই যা গোস্টস্ক্রিপ্ট দিয়ে করা যায়না।

/etc/init.d — এই ডিরেক্টরিতে থাকে সিস্টেমের ইনিশিয়ালাইজেশন স্ক্রিপ্ট, ইনিট যাদের নিয়ে কাজ করে। সিস্টেম বুট করার সময়ে, কী কী ডিমন চালু হবে, কী কী সার্ভিস চালু হবে, এই সমস্ত বলে দেয় এই ইনিশিয়ালাইজেশন স্ক্রিপ্টগুলো। এগুলো নিয়ে একটু আলোচনা এখানে করা যেত, কিন্তু ডিস্ট্রো থেকে ডিস্ট্রোয় এই স্ক্রিপ্ট গুলোর এবং তাদের কাজ করার নিয়মের কিছু পার্থক্য আছে। এটা আপনার ডিস্ট্রোর নিজের ডকুমেন্টেশন থেকে পড়ে নিন। পরের সেকশনে, কনফিগারেশন ফাইলগুলোর আলোচনার সময়ে আমরা আসব এই ডিরেক্টরির কথায়।

/etc/profile.d — ব্যাশ শেল বা অন্য কোনো শেলে লগ-ইন করার সময়ে যে স্ক্রিপ্ট গুলো চালানো হয় সেগুলো থাকে এই ডিরেক্টরিতে। কোন কোন শেল-স্ক্রিপ্ট চালানো হবে সেটা আবার সিস্টেম জানতে পারে '/etc/profile' ফাইল থেকে। আমরা আর একটু ভালো করে এটা জানব একটু বাদেই।

/etc/skel/ — প্রতিটি নতুন ইউজার বানিয়ে নেওয়ার ডিফল্ট ছকটা এখানে থাকে, আমরা আগেই বলেছি। যখনই কোনো নতুন ব্যবহারকারীকে যোগ করা হচ্ছে সিস্টেমে, এই ডিরেক্টরি থেকে কঙ্কাল বা স্কেলেটন ফাইলগুলোর কপি করে রেখে দেওয়া ওই ব্যবহারকারীর হোম ডিরেক্টরিতে। একটা গড় সিস্টেমে এর মানে হল '.alias', '.bash_profile', '.bashrc' ইত্যাদি ডটানন ফাইল। 'useradd' সংক্রান্ত অন্য কাজগুলো রুট বা সিস্টেমনিয়ন্তাকে হাতে করে করতে হয়, আগেই বলেছি।

/etc/sysconfig — নাম দেখে আন্দাজ করুন, সিস্টেমের কনফিগারেশন ফাইলগুলো থাকে এই ডিরেক্টরিতে। এর মধ্যে আবার সাবডিরেক্টরি থাকে, যাদের মধ্যে এক এক জাতের কনফিগারেশন ফাইল ভরা থাকে। যেমন 'isdn', 'network', 'scripts' ইত্যাদি। এই ডিরেক্টরির ফাইলের আলোচনাতেও আসছি আমরা, পরের সেকশনে। এই ডিরেক্টরির কনফিগারেশন ফাইল দিয়েই সিস্টেম জানতে পারে, সময় কাঠামো কী হবে, মানে কোন দেশের কোন এলাকার সময় দেখাতে হবে, বা কীবোর্ড কী হবে, ইত্যাদি। বা আইডিই ডিস্কগুলোয় ডিএমএ (DMA — **D**irect-**M**emory-**A**ccess) চালু হবে কিনা সেটাও ঠিক হয় এই ডিরেক্টরির ফাইল দিয়েই।

এই সরাসরি-স্মৃতি-সংযোগ কাকে বলে মনে আছে — যখন কোনো তথ্য বাইট পিসির মূল মেমরি বা মেমরির কোনো জায়গা আর কোনো একটা আইও যন্ত্রাংশের মধ্যে নড়াচড়া করে, এই স্থানান্তরটা ঘটে দুটো স্টেপে। স্টেপ এক, সিপিইউ বা কেন্দ্রীয় প্রসেসর ওই তথ্যের বাইটটাকে নিজের শরীরে কোনো একটা রেজিস্টারে ভরে নেয়। রেজিস্টার কাকে বলে মনে আছে — ছবি দিয়ে আমরা দেখিয়েছিলাম, দুই নম্বর দিনে? দ্বিতীয় স্টেপ হল এবার ওই বাইটটাকে তার আদত গন্তব্যে গিয়ে লিখে দেওয়া। কাজের এই ধারার দুটো ব্যথা আছে। এক, বাইট-ভ্রমণ চলাকালীন তথ্যের বাইটটাকে ফাইনালি খালাস করার আগে অব্দি সিপিইউ-র আর কিছু করার জো থাকেনা। আর দুই, এইরকম একটা এলেবেলে রকমের কুচো কাজের জন্যেও সিপিইউ-কে দুবার দুটো আলাদা কাজে লাগাতে হচ্ছে। দুপাঁচটা জনগনকে টপকে দেওয়ার জন্যেও যখন দারোগা বা ডিসি-ফিসিকে দিয়ে আর হচ্ছেনা, মাঝরাত্তিরে মুখখোমোনতিকে সাংস্কৃতিক গনতান্ত্রিক চৈতন্যবাজ ঘুম কচলে ফোন ধরে ব্যাপারটা সামলাতে হচ্ছে। এখানে একটা সাউথ দমদম, ওখানে একটা শিলিগুড়ি, এই স্কেলে কাজ চললে তবু হত, কিন্তু স্কেলটা যখন হয় জুনিয়ার সিদ্ধার্থর বরানগর বা সিনিয়র সিদ্ধার্থর ছোট-আঙারিয়া, তখন? একরাত্তিরে এক আস্তানা বাইট ইধার-উধার করার লেভেলে ব্যাপারটা সত্যিই বড্ড বেদনার হয়ে পড়ে। এইজন্যে এল ডিএমএ। এতে কাজের গতিটা অনেকটা বাড়িয়ে তোলা যায়, সিপিইউ এখন ডিএমএ কন্ট্রোলারকে বলে দেয়, ভাই আপ্ত-সহায়ক, তুমি একটু ম্যাটারটা টেকআপ করো, অমুক অমুক ঠিকানার অমুক সংখ্যক বাইটকে অমুক অমুক হিল্লে করে দাও। আইও পোর্ট থেকে মেমরি বা মেমরি থেকে আইও পোর্ট। এই ডিএমএ-ই হল সেই বস্তু যেটা নিজে থেকে চালু না হওয়ায় স্ল্যাকওয়ারকে রেডহ্যাটের মত লাগছিল, বললাম না

তখন? যাই হোক, এই '/etc/sysconfig' ডিরেক্টরির কথাতে আসছি আমরা পরের সেকশনে। আর একটা মাত্র সাবডিরেক্টরিই বাকি আছে উল্লেখ করার, এই '/etc' ডিরেক্টরিতে। তারপরেই আমরা যাব কনফিগারেশন ফাইলগুলোর আলোচনায়।

/etc/xinetd.d — এই ডিরেক্টরিতে থাকে কিছু কনফিগারেশন যাদের নিয়ে কাজ করে 'xinetd' নামে একটা ডিমন বা যখ। আগে এই কাজটা করত 'inetd' নামে অন্য একটা যখ। এই 'xinetd' হল নেট সংক্রান্ত যখ, নাম থেকেই আন্দাজ করতে পারছেন। ইন্টারনেট সংযোগগুলোর কোথায় কী ঘটছে সেটায় মনোযোগ দেওয়াই এর কাজ।

২।। গ্নু-লিনাক্সের কনফিগারেশন ফাইল

এই সেকশনটা শুরু করতে গিয়ে একটু মজাও লাগছে। আদতে এর আগে অব্দি যা যা আমরা করে এসেছি তার সমাপ্তির দিকে পৌঁছচ্ছি আমরা। এই কনফিগারেশন ফাইলগুলো, তারপরে তাদের বদলানোর তরিকা একটু আধটু, তারপরে সিস্টেমকে দিয়ে কিছু কাজ করিয়ে নেওয়ার প্রাথমিক ধাঁচের কিছু ব্যাশস্ক্রিপ্ট, তারপরেই শেষ এই জিএলটি ইশকুল পাঠমালা।

টুকরোয় টুকরোয় বারবার আলাদা আলাদা ভাবে যেসব এসেছে সেই শূন্য থেকে নয় এই দশটা দিন ধরে, তাদের একটু একসঙ্গে করে ভাবা শুরু করা যাক এবার। কনফিগারেশন ফাইল। গ্নু-লিনাক্সে কনফিগারেশন ফাইলগুলোর একটা আলাদা এবং খুব জোরালো ধরনের গুরুত্বের কথা বলেছি আমরা বারবার, '/etc/password' বা '/etc/lilo.conf' বা '/etc/fstab', এবং তাদের কোন উপাদান কী ভূমিকা পালন করে, এই উপাদান বদলে কী ভাবে বদলে দেওয়া যায় সিস্টেমের গঠন, সেটাও একটু আধটু এসেছে, বিশেষত মাউন্টের প্রসঙ্গে। এবার এদের নিয়ে সামগ্রিক আলোচনা করে নেওয়া যাক, মানে, আমাদের খোলা সুতোগুলো এবার আমরা গোটাতে শুরু করছি। যখন আমরা এই ফাইলগুলোকে তুলেছি, একটা একটা করে গাছ দেখছিলাম, কয়েকটা হাতে গোনা গাছ। এখন গোটা জঙ্গলের ভূগোলটা চিনি আমরা, '/etc' ডিরেক্টরি, মোটের উপর জঙ্গলের একটা পরিচয় নেওয়ার চেষ্টা করা যাক। এখানেও মনে রাখবেন, সম্পূর্ণ উদ্দেশ্যপ্রণোদিত রকমে, ডেলিবারেটলি, কোনো জায়গায় আমি বিশদ তালিকা দিচ্ছি না, শুধু সেইটুকু রাখছি, যাতে বিশদতায় পৌঁছনোর চেষ্টাটা কোন কোন দিক থেকে করা সম্ভব সেই হুকগুলো আপনার মাথায় তৈরি হতে পারে। এই বইটা একটা ব্যক্তিগত জার্নি — আসলে ব্যক্তিগত জার্নির একটা টেমপ্লেট, আপনি এর খুঁটিনাটিগুলো ভরাট করবেন, এই বইটা এই বই হয়ে উঠবে একমাত্র আপনার মাথাতেই। গ্নু-লিনাক্স নামে সামাজিক আন্দোলনটার ভিতর আমার ব্যক্তিগত জার্নির অভিজ্ঞতা থেকে এই টেমপ্লেটটা বানানো। তবে বানাতে গিয়ে একটা ফাটাফাটি জিনিষ হল, সত্যিই, নিজে একটা এই ধরনের ডকুমেন্টেশন তৈরি করে দেখুন, এই প্রায় দেড় লাখ শব্দ আমার লিনাক্সাধিকারের চরিত্রটাই বদলে দিল। এই চার মাসে আমার গ্নু-লিনাক্স চিন্তার রকমটাই বদলে গেল, এখন আমি আমার কম্পিউটার জানার প্রত্যেকটা জায়গাকেই আমার চিন্তার কাঠামোয় নিয়ে আসতে পারছি, যা চার মাস আগে পারতাম না, বেশ কিছু ছোট বড় গাছ ছিল, জঙ্গলটা প্রায় ছিলই-না, বা যতটুকু ছিল এখন তার চেয়ে অনেক বেশি আছে।

গ্নু-লিনাক্স একটা অপারেটিং সিস্টেম, একটা প্রোগ্রাম-সমাহার। কারনেল একটা প্রোগ্রাম, কারনেল যা যা কাজে লাগায় তারা এক একটা প্রোগ্রাম, আপনি যা চালান তা প্রোগ্রাম। আর এই সমস্ত প্রোগ্রাম চলতে গিয়ে এবং চলতে চলতে কাজে লাগায় আর বানিয়ে তোলে অনেক অনেক তথ্য। এই গোটাটা মিলিয়ে আপনার অপারেটিং সিস্টেম, আপনি একে চালান, অপারেট করেন। এই চালনীয় বা এক্সিকিউটেবল প্রোগ্রামগুলোয় থাকে অপকোড, অপারেশনাল কোড, কোনো প্রোগ্রামিং ভাষায় লেখা সোর্স কোড কম্পাইল করে যাদের পাওয়া গেছে। এই অপকোড হল সিপিইউ-র জন্যে, এদের পড়েই সে নানা কাজ করার আদেশ পায়, মানে, কম্পিউটারটা চলে। '/bin' ডিরেক্টরির 'ls' ফাইলটা একটা প্রোগ্রাম ফাইল, কমান্ড প্রম্পটে '`ls`' কমান্ড দিলেই সে চলে, মানে, '/bin/ls' ফাইলটার মধ্যে রাখা অপকোড পাঠ করে লিপিবদ্ধ আদেশটা পালন করে সিপিইউ, পালন করে আদেশের মধ্যে থাকা প্রতিটি কাজ। আমরা বর্তমান ডিরেক্টরির ফাইল তালিকা দেখতে পাই স্ক্রিনে। এই আদেশটা পালন করতে গিয়ে যে যে কাজ যে যে রকমে করে সিপিইউ তার প্রত্যেকটা রকমকেই নিয়ন্ত্রণ করা হয় বা বদলে নেওয়া যায় এই কনফিগারেশন বা সংস্থান ফাইলগুলো দিয়ে। কিন্তু এই কনফিগারেশন ফাইলগুলোর নানা ধরনের আকার এবং চরিত্র এবং গঠন। এর মধ্যে '/etc/shells' গোছের খুব সরল ফাইলও হয়, জাস্ট একটা টেক্সট ফাইলে, মধ্যে একটা তালিকা, পরপর এক একটা লাইনে যে যে

শেল এই সিস্টেমে চালানো যাবে তার তালিকা। আবার থাকে বেমক্কা রকমের জটিল সেন্ডমেল বা অ্যাপাচে সার্ভারের কনফিগারেশন ফাইল।

আমাদের কোলকাতা লাগের ঘোষিত গীক, কোডগ্রস্ত সায়মিন্দুও যে গলায় আমার মেশিনে অঙ্কুর বাংলা লাইভ সিডি দেখাতে দেখাতে বলল, 'সেন্ডমেইল, ও তো বড্ড শক্ত', এবং সঙ্কর্ষণ-ও মাথা ঘুরিয়ে আমার বইয়ের তাকে সবচেয়ে সুমো তিনখানা বই এক বিঘতের মধ্যে নিয়ে দেখাল, 'এই যে দেখো, সেন্ডমেলের ম্যানুয়ালটার সাইজ এর চেয়েও বড়', তার পর থেকেই আমার খুব ইচ্ছে করছে, যার উপর খার আছে এরকম কাউকে সেন্ডমেল কনফিগার করায় উৎসাহিত করার। উপর চালাক একজন লিনাক্সাকাঙ্খী, আজকাল তো এদের সংখ্যা বেজায় বেড়ে যাচ্ছে, আমার কাছে কোনো একটা ডিস্ট্রো চাওয়ায়, 'দিয়োতো দীপঙ্কর, কেমন করেছে দেখব' — তার গলায় এই বাড়তি শ্লেষ্মার কারণ যাদবপুরের সেই সিনিয়ারের সিটিজেনের মত, ইনিও একটা কলেজে কম্পিউটার সায়েন্স পড়ান। তথাগত আমায় বলেছিল, 'তুমি ওকে স্ল্যাকওয়ার দিয়ে দাও, বলো, এটাই সবচেয়ে সহজ, দেখছেন না নামটা — ঢিলে, স্ল্যাক'। তথাগতর এই প্রস্তাবের আভ্যন্তরীন হিংস্রতাটা আপনি বুঝতেই পারবেন না, স্ল্যাকে এক্স উইনডোজ কনফিগার করার চেষ্টা করার আগে অব্দি।

কারণ নতুন নতুন গ্নু-লিনাক্সে এসে ভারি অপরিচিত লাগে, নার্ভাস লাগে, এই বৈচিত্রটা তার মূল কারণ। কনফিগারেশন ফাইলগুলো আসে প্রোগ্রামের হাত ধরে, মূলত প্রোগ্রামার বা প্রোগ্রামারদের মর্জি মেনে, এর কোনো আলাদা নিয়ম নেই, যে প্রোগ্রামার যেভাবে চায় সেভাবেই করে। আর গ্নু-লিনাক্স তো হল একটা যোগফল, অজস্র প্রোগ্রামার, তাদের প্রোগ্রাম, তার ডেটা, তার ব্যবহার, ব্যবহারকারী — এই গোটাটা মিলিয়ে। তাই এই বৈচিত্রটা একটা অবশ্যম্ভাবী ব্যাপার। নানা ধরনের চেষ্টা চলে যদিও এই প্রোগ্রামগুলোকে এবং এর ব্যবহারগুলোকে এক ধরনের একটা একদেশতা দেওয়ার। একটা প্রোগ্রাম দেখে মনে হল এটা খায়, পরেরটা দেখে মনে হল, এটা তো মাথায় দেয়, এরকম যাতে না হয়।

এই চেষ্টাগুলো আবার নানা রকমের। এর মধ্যে একটা গুরুত্বপূর্ণ প্রচেষ্টা হল এলএসবি (LSB — **L**inux-**S**tandard-**B**ase)। গোটা নামটা না-বলে এলএসবি বলে ডাকাটাই পছন্দ করে এলএসবি সংগঠন (www.linuxbase.org), তার কারণ, শুধু গ্নু-লিনাক্স না, অন্য যে কোনো ইউনিক্স বা ইউনিক্স-ক্লোনের উপরেই প্রয়োগ করা যায় এই এলএসবি মানদণ্ড। একটা কম্পেটিবিলিটি বা স্থানান্তরযোগ্যতা মানে এক অপারেটিং সিস্টেম থেকে অন্য অপারেটিং সিস্টেমে নিয়ে যাওয়ার নিরিখ হিশেবে। এলএসবি-র উদ্দেশ্য হল নানা সিস্টেমের নানা জাতের নানা প্রকারের বাইনারির নানা ধরনের অপকোড ভাষার জায়গায় কিছু বাইনারি স্ট্যান্ডার্ড তৈরি করা, যাতে একটা বাইনারি নানা ধরনের সিস্টেমের মধ্যেই স্থানান্তরযোগ্য হতে পারে। ওই স্ট্যান্ডার্ড মেনে তৈরি বাইনারিকে এবার যে কোনো সিস্টেমেই পাঠ করে নিতে পারবে সিপিইউ এবং পালন করতে পারবে নিহিত আদেশমালা।

যাইহোক, '/etc' ডিরেক্টরি থেকে কিছু খুব প্রাত্যহিক কনফিগারেশন ফাইলের কথায় আসা যাক। আমরা এখানে অল্প কিছুকেই বেছে নিচ্ছি, এর বাইরেও অনেক কনফিগারেশন ফাইল থাকে সিস্টেমে, ব্যবহার করতে করতে আপনি নিজেই জেনে যাবেন। এই কনফিগারেশন ফাইলদের মধ্যে কিছু আবার কারনেলের কনফিগারেশন চিহ্নিত করে, তাদের বলে সিস্টেম কনফিগারেশন ফাইল। ঠিক অন্য প্রোগ্রামের যেমন, চলার প্রক্রিয়ার নানান খুঁটিনাটি ঠিক করে দিতে হয়, নইলে চলবে কী করে, কারনেলেরও তাই, কারনেল তো নিজেও একটা প্রোগ্রাম, আগেও বলেছি। কারনেলের জানার প্রয়োজন পড়ে, কী কী ইউজার আছে সিস্টেমে, কী কী গ্রুপে, বিভিন্ন ফাইল এবং ডিরেক্টরির মালিকানা এবং অনুমতির কাঠামোটা কী। কারণ, এক আর দুই নম্বর দিনে বারবার বলেছি, সিস্টেমে আপনি কাজ করছেন যখন, কাজ করছেন মানে বিভিন্ন প্রোগ্রাম চালাচ্ছেন, চলতে গিয়ে প্রোগ্রামগুলোর নানা রসদ ব্যবহার করা দরকার পড়ছে, এই রসদগুলোর ভাণ্ডারী তো কারনেল, কারনেলের কাছে দরখাস্ত পাঠাচ্ছে। কারনেলকে তো বিচার করতে হবে, সেই দরখাস্ত মঞ্জুর করা যায় কিনা। ধরুন, আপনি ইউজার হয়ে একটা সিস্টেম বাইনারি চালাতে বা বদলাতে চাইলেন, সেই দরখাস্ত সে মঞ্জুর করতে পারবে না, বা আপনি আপনার লেখা প্রবন্ধের একটা টেক্সট ফাইল এমপ্লেয়ার দিয়ে সিনেমা হিসেবে চালিয়ে দেখতে চাইলেন, আপনার চারপাশে চলমান সময়ের চলমানচিত্র কতটা উপস্থিত হয়েছে আপনার লেখায়, তখন কারনেলের কোনো আপত্তি নেই, কিন্তু আপত্তি হবে ওই এমপ্লেয়ার প্রোগ্রামের, সে তো প্রবন্ধকে সিনেমা হিশেবে দেখাতে শেখেনি। এমপ্লেয়ার প্রোগ্রামটা যারা ডিভেলপ করছে তাদের

একটা ইমেল করবেন তো পারলে, প্রোগ্রাম হিশেবে এই দারিদ্র আর কতকাল চলবে, শুধু সিনেমাকেই সিনেমা হিশেবে দেখানো?
আর ওই ইউজার হয়ে সিস্টেম বাইনারি বা সিস্টেম কনফিগারেশন বদলাতে না পারা, এমনকি বহু ফাইল দেখতেও না পারা — এইটাই কিন্তু গ্নু-লিনাক্স তথা ইউনিক্স সিস্টেমের ভাইরাস অনাক্রম্য হওয়ার কারণ। দেখুন তো নিজেই দুটোকে মিলিয়ে ভেবে নিতে পারছেন কিনা। না-হলে, দুই আর ছয় নম্বর দিন আর একবার পড়ে আসুন। যাকগে এবার কনফিগারেশন ফাইলগুলোর একটা অসমাপ্ত তালিকা রাখা যাক। এমনকি আমি যদি এইমুহূর্তে চালু প্রতিটা ডিস্ট্রোর প্রতিটি কনফিগারেশন ফাইলের তালিকাটাও এখানে তুলে দিতাম, একে তো সেটা বীভৎস বিরাট হত, কারণ ডিস্ট্রো থেকে ডিস্ট্রোয় কিছু তফাত থাকেই, আর সেটাও কিন্তু অসমাপ্তই থাকত, কারণ, যতদিনে আমার লেখা এবং আপনার পড়া হত, ততদিনে তো সেটা ইতিমধ্যেই বদলে যেত, গ্নু-লিনাক্স একটা জ্যান্ত সত্তা, অনেক জ্যান্ত প্রোগ্রাম প্রোগ্রামার আর ইউজারের। যেমন, আপনি যদি খুব সত্যবাদী হন, আপনাকে কেউ জিগেশ করলে, এখন তোমার বয়স কত, আপনি কিছুতেই উত্তর দিতে পারবেন না, তার চাওয়ার আর আপনার দেওয়ার মুহূর্তের মধ্যেও তো আপনার বয়স বদলে যাচ্ছে।

২.১।। ডিরেক্টরি '/etc/init.d' বা '/etc/rc.d' বা '/etc/rc'

এই দুটো নামের যে নামেই থাকুক না কেন ডিরেক্টরিটা, এমনকি দু একটা ডিস্ট্রোতে '/etc/rc' নামেও থাকতে পারে, এর মধ্যে থাকে সিস্টেম ইনিশিয়ালাইজেশন স্ক্রিপ্টগুলো, আগেই বলেছি। সুজে তার '/etc/init.d' ডিরেক্টরিতে দেওয়া 'README' ফাইলে লিখেছে, অনেক ডিস্ট্রোতে ব্যবহৃত '/etc/rc.d' ডিরেক্টরির জায়গায় তারা '/etc/init.d' ডিরেক্টরিতে সিস্টেমের স্টার্টআপ স্ক্রিপ্টগুলো রেখেছে ওই এলএসবি মানদণ্ডের জায়গা থেকে, একটু আগে যার কথা বললাম। স্ল্যাকওয়ারে ফাইলগুলো যেমন '/etc/rc.d' ডিরেক্টরিতেই থাকে।
গ্নু-লিনাক্স একটা সিস্টেম যখন বুট করে, পাঁচ নম্বর দিনের আলোচনাটা মনে করুন, পুরো ব্যাপারটা ঘটে পরপর কয়েকটা স্টেপে। প্রথমে বুট লোডার, আমরা লিলো দিয়ে দেখিয়েছি, সে কারনেল ইমেজটা পড়ে, পড়ে জাগিয়ে তোলে কারনেলকে। স্টেপ দুই কারনেল এবার জাগায় ইনিট নামের প্রোগ্রামকে। পাঁচ নম্বর দিনের আলোচনার সময় আমরা জানতাম না, এখন জানি, এই প্রোগ্রামটা একটা সিস্টেম বাইনারি, থাকে '/sbin' ডিরেক্টরিতে। এই '/sbin/init' প্রোগ্রাম এবার চালায় '/etc/init.d' ডিরেক্টরি থেকে পরপর স্টার্টআপ স্ক্রিপ্ট গুলোকে। ইনিট প্রোগ্রাম আবার তার নিজের কনফিগারেশন পায় '/etc/inittab' ফাইল থেকে। এই 'inittab' ফাইলের কয়েকটা লাইন একটু তুলে দিই। এতদূর যে আপনি পড়ে এসেছেন, এতেই সপ্রমাণ, আপনি পড়া ছাড়া বাঁচতে পারেন না, যা পান তাই পড়েন, আমার এক পাড়াতুতো ভাইঝির মত, সে আর কিছু না-পেলে মুগের ডালের কাগজের ঠোঙা আঠা খুলে পড়ে, মুগের ডাল ওদের অবশ্য কমই আসে, বাবার মাইনে মাত্র আঠেরোশো পঞ্চান্ন, টুম্পাও খুব ভোগে এবং সহজে চটে যায়, কিন্তু ওর মা ওকে লডেনাম খাওয়াতে চাইলেও কিনে উঠতে পারবে না, এটাই তৃতীয় বিশ্ব, এদের আডা লাভলেস হওয়াটা গোড়া থেকেই চিত্রনাট্যে নেই। দেখুন তো, লাইনগুলো পড়ে, এই মাত্র যে কথাগুলো হল, তার সঙ্গে মেলাতে পারেন কিনা।

```
# /etc/inittab
# This is the main configuration file of /etc/init, which
# is executed by the kernel on startup. It describes what
# scripts are used for the different run-levels.
# All scripts for runlevel changes are in /etc/init.d/.
# The default runlevel is defined here
id:3:initdefault:
# First script to be executed, if not booting in emergency (-b) mode
si::bootwait:/etc/init.d/boot
# /etc/init.d/rc takes care of runlevel handling
# runlevel 0  is  System halt   (Do not use this for initdefault!)
# runlevel 1  is  Single user mode
# runlevel 2  is  Local multiuser without remote network (e.g. NFS)
# runlevel 3  is  Full multiuser with network
# runlevel 4  is  Not used
# runlevel 5  is  Full multiuser with network and xdm
```

```
# runlevel 6  is  System reboot (Do not use this for initdefault!)
l0:0:wait:/etc/init.d/rc 0
l1:1:wait:/etc/init.d/rc 1
l2:2:wait:/etc/init.d/rc 2
l3:3:wait:/etc/init.d/rc 3
#l4:4:wait:/etc/init.d/rc 4
l5:5:wait:/etc/init.d/rc 5
l6:6:wait:/etc/init.d/rc 6
```

একদম উপরের লাইনটা হল ফাইলের নাম, হেডিং-এর মত করে দেওয়া, আর এই লাইন তথা পরের সবগুলো লাইনই যা যা আপনার পড়ার জন্যে তার গোড়ায় '#' চিহ্ন দেওয়া, আগেই তো বলেছি, এর মানে কমেন্ট আউট করে দেওয়া, মানবপাঠ্য করে দেওয়া। মেশিনপাঠ্য লাইনগুলোর শুরুতে এই চিহ্ন নেই। এরকম সিস্টেমের পড়ার তথা কাজ করার মত লাইন এখানে তুলে দেওয়া ২৪ লাইনের ভিতর আছে মাত্র আটটা। তাদের ভিতর প্রথমটা দেখুন, 'id:3:initdefault:' — এই লাইনটার মানে কী? এর উপরের লাইনে মানেটা দেওয়া আছে, এটাই সিস্টেমের ডিফল্ট রানলেভেল। পাঁচ নম্বর দিন থেকে রানলেভেল প্রসঙ্গ মনে করুন। বিভিন্ন রানলেভেল, তাদের আলাদা আলাদা তাৎপর্য। যদি ভালো মনে করতে না-পারেন, নো ফিয়ার, এগারো থেকে সতেরো নম্বর লাইন পড়ে মনে করে নিন। এটা আমার সিস্টেমের ডিফল্ট রানলেভেল, তিন, এর মানে পুরো মাল্টিইউজার, সব ইউজারই ব্যবহার করতে পারবে। নেটওয়ার্ক, তার মানে ইন্টারনেট সংযোগ করা যাবে, ল্যান বা ওই ধরনের কোনো সংযোগের ব্যবস্থা থাকলে তাও করা যাবে। তাহলে কী যাবেনা? বালজাক লিখতে বসতেন কোলে বিড়াল নিয়ে, সংস্কারের মত, লিখতে গেলেই বসতে হত, আপনারা অনেকে যেমন ছবির উপরে দৌড়তে থাকা ইঁদুরের পেটে কাতুকুতু না-দিয়ে কম্পিউটার করতে পারেন না — সেটা এখানে চলবে না, কিছুটা পরিমাণ ইঁদুর যদিও ব্যবহার করা যায়, জিপিএম (gpm — **G**eneral-**P**urpose-**M**ouse) বলে একটা প্যাকেজ দিয়ে আমি কনসোল মোডেই ব্যবহার করি। কিন্তু গুই বা গ্রাফিকাল ইউজার ইন্টারফেস আপনা থেকে আমার সিস্টেমে চালু হয়না। আমাকে 'startx' কমান্ড দিয়ে ঢুকতে হয় এক্স-উইনডোজে। বা, একটা স্ক্রিপ্ট বানিয়ে রেখেছি 'xpick' বলে, সেইটা দিয়ে এন্টার মারলেই সিস্টেম জানতে চায়, কোন রকমের এক্স-উইনডোজে যেতে চাই, চার রকম ব্যবস্থা আছে, প্রত্যেকটার এক একটা নম্বর, এক থেকে চার। যে নম্বরটা দিয়ে এন্টার মারি সেই ধরনের এক্স-উইনডোজ চালু হয়। দাঁড়ান, স্ক্রিপ্টটা তুলে দিই এখানে। আজকের আলোচনার শেষ দিকে আমরা আরো ভালো করে বুঝতে পারব, এই স্ক্রিপ্টটা কী ভাবে কাজ করছে। অল্প তথ্য থেকে অনেকটা আন্দাজ করার জরুরি অভ্যেশটা করতে শুরু করুন, গরীব দেশের গরীব মানুষের অত পড়াশুনো করার পয়সা কোথায়?

```
#!/bin/bash
echo "Choose a number to pick your Window Manager"
echo ""
echo "1.  KDE"
echo ""
echo "2. GNOME"
echo ""
echo "3. FLUXBOX"
echo ""
echo "4. BLACKBOX"
echo ""
read NUMBER
case $NUMBER in
   1) export WINDOWMANAGER=startkde ;;
   2) export WINDOWMANAGER=gnome-session ;;
   3) export WINDOWMANAGER=fluxbox ;;
   4) export WINDOWMANAGER=blackbox ;;
   *) echo ""; echo "Are You Literate? Try Again." ; echo "" ; exit ;;
esac
exec startx
```

একটা জিনিষ দেখুন — এর শেষ লাইনটা, তাতে বলা আছে '`startx`' কমান্ডটাই এক্সিকিউট করতে মানে চালাতে। তার আগে শুধু ব্যবস্থা করে রাখা হয়েছে যাতে কনফিগারেশন বদলে '`KDE`' বা '`Gnome`' বা '`Fluxbox`' বা

'Blackbox' যেটাই চাই সেই রকম এক্স-উইনডোজে যাতে যেতে পারি। এই নামগুলো তো ইতিমধ্যেই আপনার চেনা। এই স্ক্রিপ্ট টার প্রথম লাইনটা দেখুন, সেই '#!' চিহ্ন দিয়েই শুরু, কিন্তু এই লাইনটা সিস্টেমের পাঠ্য। আসলে এখানে একটা চিহ্ন নেই, আছে দুটো চিহ্ন, '#' আর '!'। দুটোকে একসঙ্গে আদর করে ডাকা হয় 'শা-ব্যাং' বলে। 'শা'-টা আসছে '#' চিহ্নের ইউরোপিয় নাম 'আশ' থেকে। আর '!' চিহ্নটা কেন 'ব্যাং' মানে সাহেবী বন্দুকের গুলির শব্দ, দেশি বন্দুকে যেমন 'দুম' শব্দ হয়, সেটা আপনার বাড়ির কুচো কাউকে জিগেশ করে নিন, যে কমিকস পড়ে। এই শা-ব্যাং আসলে সিস্টেমকে বলে দেয় যে এই স্ক্রিপ্টের পরের লাইনগুলো পড়তে হবে 'bash' নামক প্রোগ্রামের নিয়ম মেনে। যাকগে এটায় আমরা পরে আসছি। এই প্রোগ্রামটা থাকে কোথায় বা দিনের ঠিক কোন সময়ে এটা মাথায় দিতে হয়, এসব আপনি জানেন নাকি?

হ্যাঁ, যা বলছিলাম, এই '/etc/inittab' ফাইলটাতেই দেওয়া আছে দেখুন, রানলেভেল পাঁচ মানে ফুল মাল্টিইউজার এবং নেটওয়ার্ক, ঠিক রানলেভেল তিনের মতই, এবং তার সঙ্গে আরো কিছু, 'and xdm' — সেই ছবি-ইঁদুর মানে গুই চালানোর আদেশ, যখন ডিফল্টটাই গুই, মানে, ঠিক মাইক্রোসফট উইনডোজে যেমন, মেশিনটা চালুই হবে গুইতে। এক্সডিএম মানে এক্স-উইনডোজ ডিসপ্লে ম্যানেজার। প্রত্যেকটা ডিস্ট্রিবিউশনেই আলাদা আলাদা ব্যবস্থা থাকে, এক রানলেভেল থেকে আর এক রানলেভেলে বদলানোর, বা একটা বিশেষ রানলেভেলে কী কী সার্ভিস চালু হবে, সেটা বদলানোর। যেমন, রানলেভেল তিন থেকে পাঁচে বদলানোর জন্যে 'xdm' নামে সার্ভিসটা বাড়াতে হচ্ছে। এছাড়া, ইনস্টল করার সময়েও জিগেশ করে নেয়, কী কী সার্ভিস চালু হবে, তাতে আপনার কোনো আলাদা আলাদা পছন্দ আছে কিনা। কিন্তু সেই সমস্তই আদতে যা করে তাহলে পর্দার পিছনে রয়ে যাওয়া '/etc/inittab' ফাইল এবং '/etc/init.d' ডিরেক্টরির স্ক্রিপ্টগুলোকে বদলানো। ধরুন আপনার মেশিন চালু হয় কমান্ড প্রম্পটে, এই মেশিনের মত, এবার চাইছেন ইঁদুরের পেটে কাতুকুতু দিয়েই আপনার যন্ত্রগণনা শুরু করতে, তখন কী করবেন?

যাকগে, রানলেভেলটা বদলাতে হলে সবচেয়ে সহজ উপায় হল 'emacs /etc/inittab' কমান্ড দিয়ে ফাইলটা খোলা, তারপর জাস্ট 'id:3:initdefault:'-এর '3' উড়িয়ে '5' করে দেওয়া। তারপর কন্ট্রোল-এক্স কন্ট্রোল-এস মেরে সেভ করা, এবং কন্ট্রোল-এক্স কন্ট্রোল-সি মেরে বেরিয়ে আসা। পরের বার যখন মেশিন বুট করবে, দেখবেন, আরিববাস, পুরো গুই, আপনার স্ক্রিন কী লাগবে, কম্পিউটার তো না, পুরো সিনেমা। তবে এই গোটা কাজটা করার আগে 'su' মেরে এবং রুট পাসওয়ার্ড দিয়ে রুট হয়ে নেবেন, নইলে কিন্তু ফাইলটা খুলবে রিড-অনলি রকমে, সেভ করতে দেবেনা। '/etc' ডিরেক্টরির এবং তার ভিতরকার সমস্ত সাবডিরেক্টরি এবং ফাইলের একচ্ছত্র মালিকানা সিস্টেমনিয়ন্তা রুটের, অন্য কোনো ইউজারের কোনো অধিকার নেই এখানকার কোনো ফাইল বদলানোর। এখন '/etc/inittab' বদলানোটা দেখলেন, এর আগে দেখেছেন '/etc/fstab' ফাইল বদলানো, এরকম সমস্ত কনফিগারেশন ফাইলই তাই — ছয় নম্বর দিনে বলেছিলাম না, গোটা গ্নু-লিনাক্স সিস্টেমের প্রতিটি কিছুকে আপনি নিয়ন্ত্রণ করতে পারেন, নিজের পছন্দমাফিক। এইরকম ভাবে বদলানো যায় ওই স্ক্রিপ্টগুলোকেও, '/etc/init.d' ডিরেক্টরির ভিতরকার যে ডিরেক্টরিগুলোর কথা এখানে উল্লিখিত আছে, প্রত্যেকটাতেই আছে এক এক রকম স্ক্রিপ্ট।

'/etc/init.d' ডিরেক্টরির ভিতরকার ফাইল আর ডিরেক্টরিগুলোর একটা তালিকা দেওয়া যাক। পাশে 'd' মানে ডিরেক্টরি আর 'f' মানে ফাইল। এই তালিকাটা কিন্তু আদৌ সমাপ্ত না, আমি কিছু কিছু ফাইল এখানে তুলে দিলাম, যেগুলো এর আগে আমাদের আলোচনায় এসেছে, বা পরে আসতে পারে, বা সাধারণ একটা বাড়ির পিসিতে যে যে ফাইল বোঝার দরকার পড়তে পারে, বা যাদের নাম দেখে আপনার কিছু আন্দাজ আসতে পারে, এই রকম। এর বাইরেও কিছু ফাইল রয়ে গেল।

/etc/init.d ডিরেক্টরির মধ্যের ডিরেক্টরি (d) এবং ফাইলের (f) তালিকা											
d	boot.d	d	rc4.d	f	boot	f	network	f	rc	f	smpppd
d	init.d	d	rc5.d	f	cron	f	nfs	f	resmgr	f	splash
d	rc0.d	d	rc6.d	f	fbset	f	portmap	f	rpasswdd	f	syslog
d	rc1.d	f	alsasound	f	gpm	f	postfix	f	setserial	f	xdm
d	rc2.d	f	atd	f	halt	f	random	f	single	f	xfs
d	rc3.d	f	autofs	f	kbd	f	raw	f	skeleton	f	xinetd

২.২।। ডিরেক্টরি '/etc/sysconfig'

আমরা '/etc' ডিরেক্টরি আলোচনার সময় যে দুটো ডিরেক্টরির কথা আলাদা করে বলেছিলাম, তার মধ্যে একটা '/etc/init.d' তো চুকলো, দ্বিতীয়টা হল '/etc/sysconfig'। এই ডিরেক্টরির মধ্যেকার ফাইলগুলোর ওই একই ভাবে একটা নির্বাচিত তালিকা বানানো যাক, ডিরেক্টরিটার নাম থেকেই আন্দাজ করতে পারছেন, কী জাতের ফাইল। কোনো কোনো নামের ফাইল দেখবেন, 'init.d' ডিরেক্টরিতেও আছে, আবার 'sysconfig' ডিরেক্টরিতেও আছে। যেমন 'boot' বা 'cron' বা 'postfix' ইত্যাদি। এখানে দুটো ডিরেক্টরির মধ্যে ভূমিকাগত তফাতটা খেয়াল করুন, একটা লাগছে সিস্টেম বুট করার সময়ে, অন্যটা লাগছে একটা জ্যান্ত চালু থাকা সিস্টেমের ভিতর। দুটো ডিরেক্টরির একই নামের ফাইল দেখবেন গঠনে আলাদা, 'init.d' ডিরেক্টরির ফাইল সচরাচর 'sysconfig' ডিরেক্টরির ফাইল থেকে বড় সাইজের, এবং খুলে গোড়াটা একটু পড়ে দেখুন, সিস্টেমে এদের ভূমিকার তফাতটা কিছুটা অব্দি বুঝতে পারবেন।

/etc/sysconfig ডিরেক্টরিতে ডিরেক্টরি (d) এবং ফাইলের (f) তালিকা							
d	isdn	f	cron	f	language	f	nfs
d	network	f	cups	f	locate	f	postfix
d	scripts	f	hardware	f	lvm	f	printer
f	boot	f	ispell	f	mail	f	proxy
f	clock	f	kernel	f	mouse	f	sound
f	console	f	keyboard	f	network	f	tetex

এবার '/etc/sysconfig' থেকে দু-একটা ফাইল একটু খুলে দেখি আসুন। খোলার জন্যে আমাদের 'less' বা 'more' তো আছেই। একটা কাজ করতে পারেন, গোটা ডিরেক্টরির সবকটা ফাইলকে একটা ফাইলে কপি করে নিতে পারেন। একটা ডিরেক্টরির যাবতীয় ফাইল ধরে আপনার পছন্দমত নামের একটা ফাইলে কপি করে ফেলার জন্যে আসুন, একটা ব্যাশ স্ক্রিপ্ট বানাই এবং বুঝি। আপনি তো 'emacs' দিয়ে ফাইল লিখতে জানেন। '`emacs writefiles`' কমান্ড দিয়ে একটা ফাইল খুলুন। সেখানে নিচের এই লাইন কটা টাইপ করে দিন। অন্য যে কোনো নাম দিয়েও করতে পারেন।

```
#!/bin/bash
echo Name of file?
read f
c=1
d=********************************************
for i in *
    do echo -e "\n\n$d\n$c. $i\n" >> $f
    cat $i>> $f
    c=`expr $c + 1`
done
```

এবার, '<Ctrl><S>' মানে সেভ করে, '<Ctrl><X>' মানে বেরিয়ে আসুন। একটা জিনিষ তথাগত ধরিয়ে দিয়েছে, এখানে আমি 'S' বা 'X' লিখছি, 's' বা 'x' না লিখে তার মানে এই নয় যে আপনি শিফট টিপে ক্যাপিটাল কেস অক্ষরটা টিপবেন, আসলে কিবোর্ডে দেখুন, অক্ষরের নামগুলো সব ক্যাপিটাল কেসেই আছে। আপনি কন্ট্রোল সুইচটা টিপে রেখে এক্স সুইচটা টিপবেন। এবার '`ls`' মারুন, আপনার বানানো 'writefiles' ফাইল জ্বলজ্বল করে বিরাজ করছে ডিরেক্টরির মধ্যে। এবার আসছে স্ক্রিপ্ট বানানোর দ্বিতীয় স্টেপ, মানে এই পড়নীয় ফাইলকে চালনীয় বানানো, টেক্সট ফাইলটাকে এক্সিকিউটেবল করা। তার উপায় 'chmod' — যা করলে একটা ফাইল এক্সিকিউটেবল হয়, তার দীর্ঘ তালিকায় মানে লং লিস্টিং-এ একটা 'x' যোগ হয়। সাত নম্বর দিনের ৮.২ নম্বর সেকশনে আমরা ফাইলের অনুমতি এবং অধিকার বদলানো নিয়ে আলোচনা করেছিলাম, মনে করুন। তার মানে এবার দুভাবে এটা করা যায়, হয় '`chmod +x writefiles`' অথবা '`chmod 755 writefiles`', দুটো কমান্ডের যে কোনোটাই ফাইলটাকে এক্সিকিউটেবল বা চালনীয় করে দেবে। এর যে কোনোটা করার পরেই '`ls -l writefiles`' মেরে দেখুন, অনুমতির কাঠামো দেখাবে, '`-rwxr-xr-x`'। এবং লং লিস্টিং-এ যেমন, তারপর যথারীতি ফাইলটার লিংকের সংখ্যা, ব্যক্তি-মালিকানা এবং গ্রুপ-মালিকানা দেখাবে, আয়তন তারিখ এবং সময় দেখাবে, আর শেষে

ফাইলটার নাম। এবার যেই এক্সিকিউটেবল হয়ে গেল, অমনি আপনি চালাতে পারবেন। যে ডিরেক্টরিতে রেখেছেন স্ক্রিপ্টটা, সেই পথনির্দেশসহ গোটাটা দিয়ে চালাবেন। বা ওই ডিরেক্টরিতেই থাকলে দেবেন কমান্ড দেবেন '`./writefiles`'। এই বিন্দু বা ডট বা '`.`' মানে এখানে কারেন্ট বা বর্তমান ডিরেক্টরি। গ্নু-লিনাক্সে বর্তমান ডিরেক্টরি '$PATH'-এ থাকেনা। কেন, তার একটা মজার গল্প আছে, কিন্তু ছেড়ে দিন, আর বেশি বাড়ানো যাবেনা বইটা। আরো এইসব বাজে বকা দিয়ে। আমি আমার স্ক্রিপ্টগুলো রাখি আমার হোম ডিরেক্টরির মধ্যে বাইনারি ডিরেক্টরিতে, মানে '/home/dd/bin' ডিরেক্টরিতে। তাই আমাকে আর পথ দিতে হয়না, কারণ, আগেই তো লিখেছি, এই ডিরেক্টরিটা ইউজারের '$PATH'-এ থাকে। এবার স্ক্রিপ্টটা চালিয়ে দেখুন, একটা নাম জানতে চাইবে, কী নামের ফাইলে সে রাখবে এই ডিরেক্টরির যাবতীয় ফাইল, সেই নামটা আপনি দেবেন। তারপর, নির্ভুল একটার পর একটা ফাইল তুলবে আর সেখানে পরপর নম্বর দিয়ে হেডিং বানিয়ে রেখে যাবে, চালিয়ে দেখুন।

এমন যদি হয় আপনি অন্য কারুর বানানো একটা স্ক্রিপ্ট দেখছেন তখন সেই ব্যবহারকারীর বিষয়ে জানতে পারবেন 'id' বা 'finger' কমান্ড দিয়ে। যেমন আমার সুজে সিস্টেমে ব্যবহারকারীর নাম দেখাচ্ছিল 'dd' এবং গ্রুপের নাম দেখাচ্ছিল 'users'। এবার, রুট হয়ে '`id dd`' দিয়ে জানতে পারলাম 'dd' নামক ব্যবহারকারীর পরিচিতি বা '`uid=500(dd)`' এবং '`gid=100(users)`'। এবং যে যে গ্রুপের সভ্য এই 'dd' তারা হল '`groups=100(users), 14(uucp), 16(dialout), 17(audio), 20(cdrom), 33(video)`'। আর '`finger dd`' দিয়ে পেলাম —

```
Login: dd                Name: dipankar das
Directory: /home/dd      Shell: /bin/bash
Last login Sat Feb 14 10:05 (IST) on tty1
No Mail.
```

এই ইউজার পরিচিতির কমান্ড দুটো সম্পর্কে আপনার যা যা জানার আছে, আমাকে বিরক্ত করবেন না, কালকে রাত্তেই বনগাঁ লোকালে একজন সিনিয়র সিটিজেন জানাচ্ছিলেন, আমরা সবাই সমাজবিমুখ হয়ে যাচ্ছি টিভি দেখে দেখে, তাই যান, সামাজিক হোন, মানুষের সঙ্গে মিশুন, ম্যান-কে জিগেশ করুন, এই কমান্ড দুটো কী?

ফেরত আসা যাক আমাদের লেখা স্ক্রিপ্ট 'writefiles'-এর কথায়। এর এক নম্বর লাইনটা তো আমাদের চেনা, '`#!/bin/bash`'। এর মানে ব্যাশ নামে বাইনারি ফাইল দিয়ে এই স্ক্রিপ্টটাকে চালাবে সিস্টেম। এর পরের লাইনে ব্যবহার করা আছে '`echo`' নামে একটা কমান্ড, যা ইকো করে, মানে আপনি যা দেবেন তাই স্ক্রিনে ফুটিয়ে তুলবে। একটা স্ক্রিপ্ট মানে আগেই বলেছি পরপর লাইনগুলো পরপর কমান্ডের সমাহার। তাই প্রথম কমান্ডটা অনুযায়ী এবার সিস্টেম স্ক্রিনে ফুটিয়ে তোলে '`Name of file?`'। মানে, এই স্ক্রিপ্টটা আমরা বানিয়েছিলাম একটা ডিরেক্টরির পরপর সমস্ত ফাইল একটা ফাইলে, পরপর নম্বর দিয়ে কপি করে যাওয়ার জন্যে। সেটা কী নামের ফাইলে করবে সেটাই জানতে চাইছে। এখানে আপনি একটু ম্যানলি হোন এবং প্রতিধ্বনিত হওয়ার চেষ্টা করুন, কারণ এর পরেই এই 'echo'-কে আমরা একটা বিশেষ অপশন সহ ব্যবহার করব।

এর পরের লাইনে আছে '`read f`'। যে নামটা এবার আপনি টাইপ করে দিলেন সিস্টেমে, সেটাকে পড়ে ফেলার এবং সেই পড়ে-ফেলা তথ্যটুকু সিস্টেমের স্মৃতির খাতার কোনো পাতায় '`f`' নামে একটা জায়গা বানিয়ে লিখে ফেলার কথা বলা হল। এরপর থেকে যাতে '`f`'-বোলে-তো ওই নামটা বুঝে যায় সিস্টেম। এবং যে সব ফাইল গুলোকে পরপর লিখে যে ফাইলটা সে বানাবে তার নাম দেয় ওই '`f`'-নামে জায়গাটায় যা তুলে রাখা হয়েছিল তাই দিয়ে। চতুর্থ লাইনে একই কায়দায় বলা হল '`c`' নামের একটা জায়গা বানাতে এবং সেখানে '`1`' এই মানটা লিখে রাখতে। এই '`c`' একটা কাউন্টার, প্রত্যেকটা নতুন ফাইল লিখে ফেলা হবে আর এই কাউন্টারটা এক এক করে বাড়বে। পাঁচ নম্বর লাইনে আবার সেই গল্প, এখানে স্মৃতির খাতার জায়গাটার নাম '`d`', এবং এখানে মানটা হল একটা দীর্ঘ চিহ্ন-সমাহার, পঁয়তাল্লিশটা '`*`' চিহ্নের। পরপর ফাইল লিখে যাওয়ার সময় যে সমাহারটা একটা ডিভাইডর বা পাঁচিলের মত একটা ফাইল থেকে আর একটা ফাইলকে আলাদা করবে। এই '`f`', '`c`' আর '`d`' — এদের শেল স্ক্রিপ্টে বা মোটামুটি যে কোনো প্রোগ্রামিং ভাষাতেই ডাকা হয় ভ্যারিয়েবল বলে। একটা চলরাশি বা ভ্যারিয়েবল চলে বেড়ায়, এক এক পরিস্থিতিতে তার এক একটা মান ঘটে। এর বিপরীত হল স্থির রাশি বা কনস্টান্ট, যাদের মান

সবসময়েই স্থির। ধরুন, π-এর মান, বা আলোর গতি, বা কতবার আপনি মরতে পারেন, ইত্যাদি। এরা প্রাকৃতিক স্থির রাশি, আর প্রোগ্রামিং-এর বেলায় তারাই কন্সটান্ট যাদের মান ওই প্রোগ্রামের মধ্যে বদলাবে না।

ছয় নম্বর লাইন থেকে শুরু একটা লুপ, তিন নম্বর দিনে আডার প্রসঙ্গে লুপের আলোচনা মনে আছে? ঠিক সেইরকম এখানে '`for i in *`' বাক্যাংশ বোঝাচ্ছে যে ওই ডিরেক্টরিতে মোট যত ফাইল আছে তাদের সবাইকে আমরা বুঝছি '`*`' দিয়ে, তারপর '`i`' নামে একটা ভ্যারিয়েবল তৈরি করছি, এবং একটা একটা করে ফাইল পড়তে থাকছি। ধরুন সেখানে যদি তিনটে ফাইল থাকে, যাদের নাম '`onefile`', '`twofile`' আর '`threefile`', তাহলে প্রথমবার '`i`' ভ্যারিয়েবল-এর ভ্যালু হবে প্রথম ফাইলের নাম, মানে '`onefile`'। '`i`' নামের চলরাশির মান — এই গোটা কথাটাকে ছোট করে লেখা হয় '`$i`', তার মানে এই সময় '`$i`' সমান '`onefile`'। এই মান নেওয়া মাত্র প্রথম লুপটা চালু হবে। পরেরবার আবার যেই লুপের শুরুতে আমরা আসব, লুপ মানেই তো তাই, ফিরে আসা বারবার, একই জায়গায়, তখন কিন্তু '`i`'-এর মান মানে '`$i`' হবে দ্বিতীয় ফাইলের নাম মানে '`twofile`'। তৃতীয়বার '`$i`' হবে '`threefile`'। এখানে তিনবারের পরই লুপ খতম হয়ে যাবে, কারন এখানে ফাইলের সংখ্যা তিন। ফাইল বেশি হলে লুপও ঘুরবে বেশিবার। লুপের প্রতিবার আবর্তনে একটা একটা করে ফাইলের নাম তুলে নেবে '`i`'। ছয় নম্বর লাইনে, '`for i in *`' লুপটা শুরু, এবং দশ নম্বর লাইনে, '`done`', লুপটা শেষ।

লুপের মধ্যের তিনটে লাইনে কী আছে? মানে এক একবার এক একটা করে ফাইলের নাম নিয়ে ব্যাশ কী করবে — সেটাই লেখা আছে। ধরে নিন, আমাদের এই স্ক্রিপ্ট আমাদের এই মাত্র দেওয়া তিনটে ফাইলের উদাহরণটা নিয়েই কাজ করছে। এবং লুপের প্রথম আবর্তন চলছে। তাহলে আমরা জানি '`f`'-এর মান, মানে '`$f`'। ফাইলের নাম যা আমরা দিয়েছি, যে নামের ফাইলে এই ডিরেক্টরির যাবতীয় ফাইলকে লিখে ফেলতে বলেছি আমরা। ধরুন ব্যাশ আমাদের কাছে '`Name of file?`' জিগেশ করার পর আমরা দিয়েছি '`all.files`', তাহলে '`f`'-এর মান মানে '`$f`' হল '`all.files`'। আমরা জানি '`c`'-এর মান, মানে '`$c`' এখন এক বা '`1`', সেই মানটা আমরাই দিয়ে দিয়েছি স্ক্রিপ্ট-এর চার নম্বর লাইনে। এবং '`d`'-এর মান মানে '`$d`', অর্থাৎ ডিভাইডর-এর আকারটাও জানি আমরা, পঁয়তাল্লিশ পিস পঞ্চমুখ তারকা, সেটাও আমরাই দিয়েছি। এবং এই মুহূর্তে '`i`'-এর মান মানে '`$i`' হল '`onefile`', ডিরেক্টরি থেকে পড়া হয়েছে। এই '`$i`' আর '`$c`' লুপের প্রতিটি আবর্তনেই বদলে যেতে থাকবে। কিন্তু '`$f`' আর '`$d`' একই থাকবে। দেখা যাক। চারটে ভ্যারিয়েবলের এই এই মান নিয়ে লুপ শুরু হল। ছয় নম্বর লাইনে।

সাত নম্বর লাইনে দেখুন, একটা বিশেষ অপশন সহ, '`-e`', ইকো করতে বলা হয়েছে ডাবল কোটের মধ্যে রাখা '`\n\n$d\n$c. $i\n`' অংশটাকে। এই ডাবল কোটের আর সিংগল কোটের সবিশেষ সব চককর আছে, পরে আসব আমরা। এখন জাস্ট মাথায় রাখুন, এর মধ্যে '`\n`' অংশটা হল নিউলাইন, মানে নতুন লাইন শুরু করতে বলা, মানে এক লাইন জায়গা বাদ দিতে বলা। ওই '`-e`' অপশন আসলে এইভাবে পড়তে বলে ইকোকে, নইলে ইকো একদম আক্ষরিক ভাবে ফুটিয়ে তুলত '`\n`', এক লাইন জায়গা ছাড়ার বদলে। এবার তাহলে ডাবল কোটের মধ্যে ওই গোটাটা কী হয়ে দাঁড়াল? দেখুন, দুটো লাইন ফাঁকা দাও, তারপরে '`d`' ভ্যারিয়েবল-এর মানটাকে লেখো, মানে সেই তারার পাঁচিল, তারপরে আবার একটা লাইন ফাঁকা দাও, তারপরে লেখো '`c`'-এর মান, যা এই মুহূর্তে '`1`'। তারপরে একটা বিন্দুচিহ্ন দাও, তারপরে, একটা স্পেস বা ফাঁকা জায়গা দাও, তারপর '`i`'-এর মান লেখো, মানে ডিরেক্টরি থেকে পড়া ফাইলের নাম, মানে এই মুহর্তে সেটা '`onefile`'। তারপরে আবার এক লাইন ফাঁকা দাও। এবার এই গোটাটাকে লেখা হল মানে রিডাইরেক্ট বা চালান করা হল কোথায়? '`$f`'-এ, মানে যা এখন '`all.files`'। দু লাইন ফাঁকা, তিন নম্বর লাইনে তারার পাঁচিল, তার নিচে এক লাইন ফাঁকা, পাঁচ নম্বর লাইনে এল '`1. onefile`', তারপর আবার এক লাইন ফাঁকা — এই ছটা লাইন গোটাটাই লিখে ফেলা হল '`all.files`' ফাইলে, যা বানিয়ে চলছে স্ক্রিপ্টটা। এখানেই স্ক্রিপ্টের সাত নম্বর লাইন শেষ।

এবার আট নম্বর লাইন। দেখুন এখানেও সেই ক্যাট করে চালান করা '`all.files`' ফাইলে। এবং চালান করা হচ্ছে কাকে? গোটা '`onefile`' ফাইলটাকে ক্যাট করে দেওয়া হচ্ছে, এই কায়দাটার সঙ্গে তো আমরা আগে থেকেই পরিচিত। তার মানে একটা পাঁচিল, একটা হেডিং, তাতে ফাইলের নামটা লেখা হচ্ছে, তারপরে গোটা ফাইলটা। এখানেই শেষ হচ্ছে লুপ-এর প্রথম আবর্তন। মানে, প্রথম ফাইল লটকে দেওয়ার কাজ। এর পরের মানে,

স্ক্রিপ্টের নয় নম্বর লাইনে আছে একটা যোগের হিশেব। ম্যান করে 'expr' পড়ে ফেলুন। এই লাইনে আমরা 'c' ভ্যারিয়েবলের মান বদল করছি। প্রোগ্রামিং ভাষায় '=' চিহ্নকে বলে অ্যাসাইনমেন্ট বা ধার্যীকরণ। '=' চিহ্নের বাঁদিকে থাকে সেই ভূমি বা ভ্যারিয়েবলের নাম যেখানে আমরা মানটা রাখব, আর ডানদিকে থাকে যাকে রাখব। যেমন দেখুন এখানে আমরা কী করছি? এই মুহূর্তে '$c' হল '1'। '' চিহ্নের ডানদিকে আমরা সেই মানটার সঙ্গে এক যোগ করছি। অর্থাৎ যোগের শেষে ডানদিক হল '2'। এবার 'c' নামক স্মৃতি-র জমিতে আমরা পুরোনো মান '$c' মানে '1' মুছে লিখে দিলাম '2', এখন '$c' হয়ে দাঁড়াল '2'।

এবার 'done' মানে লুপের এই আবর্তন থেকে টাটা, গোটা স্ক্রিপ্টটার থেকেই টাটা হয়ে যেত, কিন্তু ফাইল তো এখনো শেষ হয়নি, এই 'for' লুপের শুরুর লাইনে, মানে স্ক্রিপ্টের ছয় নম্বর লাইনে ওই যে আমরা লিখে দিয়েছিলাম 'i in *'। এবার, এই উদাহরণে আমাদের যেতে হবে দ্বিতীয় ফাইলে, মানে এখন '$i' হবে 'twofile'। দেখুন, '$c' আর '$i' বদলে গেল। অথচ '$f' আর '$d' কিন্তু একই আছে, সেই একই লিখে ফেলার ফাইল 'all.files' আর সেই একই পাঁচিল। আবার লুপ চালু হল সাত নম্বর লাইন থেকে। এবার একই ভাবে পাঁচিল আর হেডিং লিখে ফেলা হল। একই ভাবে লেখা হল দ্বিতীয় ফাইলের নাম। শুধু এখন '$c' আর '$i' বদলে গেছে, তাই, আগের হেডিং-এর '1. onefile'-এর জায়গায় লেখা হল '2. twofile', এবং তারপরে 'all.files' ফাইলে রিডাইরেক্ট করে দেওয়া হল দ্বিতীয় ফাইল 'twofile'-কে। এই ভাবে লুপটা ঘুরে চলবে, যতক্ষণ না ডাইরেক্টরির সমস্ত ফাইল শেষ হচ্ছে। এখানে তিনবার ঘুরবে লুপটা। তৃতীয়বারেও পাঁচিল একই থাকবে, লিখে ফেলার ফাইল 'all.files' একই থাকবে, শুধু হেডিংটা বদলে হবে '3. threefile', এবং তার পরে ক্যাট করা হবে 'threefile' ফাইলটাকে।

আমরা বেমক্কা রকমের দীর্ঘ একটা কুপথ সেরে এলাম। চলছিল '/etc/sysconfig' ডিরেক্টরির কথা। তার সব ফাইলগুলো একসঙ্গে পড়ার জন্যে একটা টেক্সটফাইল লেখার উপায় হিশেবে স্ক্রিপ্টটা এলো, তারপর সেটার আলোচনায় এতটা। আসলে মাথায় আসছে তো, একটু বাদেই শেল স্ক্রিপ্টের কথায় আসব, তারই একটা পরিচিতি বানিয়ে রাখতে চাইছি। তবে, একটা কথা বলি, স্ক্রিপ্ট হিশেবে এটা কিন্তু অত্যন্ত খাজা হতে বাধ্য, কারণ এটা আমার বানানো, আমার কম্পিউটার জানাটা আপনারা জানেন। গতকাল রাতেই আমাদের লাগের সবচেয়ে পণ্ডিতদের একজন, মানস লাহাকে এই বইটা একটা সিডিতে পুড়িয়ে পাঠানোর কথা পাকা হল ইমেলে। এইসব স্ক্রিপ্ট দেখে মানসবাবু সিডিতে না পুড়িয়ে গোটা মালটাকেই পুড়িয়ে ফেলতে বলার একটা ম্যাসিভ চানস আছে। কিন্তু এই খাজাত্বটা এখানে একটা রাজাত্বও দিয়েছে, আপনাদের অনেকের কাছেই এটা অনেক সহজবোধ্য হবে, অন্য অনেক সত্যিকারের ভালো স্ক্রিপ্টের চেয়ে। আমি আমার জার্নি দিয়ে আমার গ্নু-লিনাক্স ভাবার রকমটা যেমন করে আপনার কাছে পৌঁছে দিতে চাইছি।

আমি আমার সুজে সিস্টেমে '/etc/sysconfig' ডিরেক্টরিতে এই স্ক্রিপ্টটা দিয়েই সমস্ত ফাইলের একটা গণকবর বানিয়েছি, যার নাম দিয়েছিলাম 'dir.sysconfig.textfile', ওই যে যখন আমার কাছে জানতে চাইল, কী নাম দেব। এই 'dir.sysconfig.textfile' ফাইলটার সাইজ ৯৪ কেবি, এবং এতে মোট ফাইল আছে ৫৩-টা। শেষটার হেডিং বানিয়ে দিয়েছে ওই স্ক্রিপ্টটা, তারার পাঁচিলের নিচে '53. ypbind'। ফাইলকবর একই ভাবে বানিয়েছিলাম '/etc/init.d' ডিরেক্টরিতেও। তার নাম 'dir.init.d.textfile'। সেটার মোট সাইজ ২৫২ কেবি। সেটাতেও শেষ ফাইল 'ypbind', কিন্তু সেখানে হেডিংটা হল '82. ypbind', মানে এখানে ফাইল আছে ৮২-টা। '/etc/init.d' ডিরেক্টরির ফাইলগুলো কত বেশি বড় '/etc/sysconfig' ডিরেক্টরির ফাইলগুলোর থেকে তার একটা আন্দাজ করতে পারছেন। এবার এই 'dir.sysconfig.textfile' ফাইল থেকে দু-একটা ফাইল কয়েক লাইন করে পড়ি আসুন। প্রথমে '/etc/sysconfig/clock' ফাইলের এন্ট্রিটা তুলে দি, এতে আপনি আমাদের 'dir.sysconfig.textfile' ফাইলের কাঠামোটাও দেখতে পাবেন সরাসরি। এটা গণকবরের দশ নম্বর ফাইল,

```
*********************************************
10. clock

## Path:      System/Environment/Clock
## Description:
## Type:      string
```

```
# Set to "-u" if your system clock is set to UTC, and to "--localtime"
# if your clock runs that way.
HWCLOCK="--localtime"
## Type:    string(Europe/Berlin,Europe/London,Europe/Paris)
#
# Timezone (e.g. CET)
# (this will set /usr/lib/zoneinfo/localtime)
TIMEZONE="Asia/Calcutta"
DEFAULT_TIMEZONE="US/Pacific"
```

এই ফাইলটার অনেকগুলো লাইনই কিন্তু মানবপাঠ্য, দেখুন, '#' চিহ্ন দিয়ে শুরু। এই যে লোকালটাইম আর টাইমজোন এই দুটো মনে করতে পারছেন, ইনস্টলেশনের সময় আপনার কাছে জানতে চেয়েছিল। একটা হল, যে সিস্টেমের ঘড়িটা কিসের সঙ্গে মেলাবে, গ্লোবাল টাইম মানে জিএমটি বা গ্রিনউইচ মিন টাইম, না লোকাল টাইম। আর অন্যটা হল, লোকাল টাইমের জোন কোনটা হবে, মানে কোন ভৌগোলিক সংস্থানের লোকাল টাইম হবে। কারণ, দেশ থেকে দেশে, ভূগোল থেকে ভূগোলে সময় তো আলাদা। ওই '#' চিহ্ন দেওয়া লাইনগুলোই সেইগুলোই বলে দিচ্ছে। যেমন আমার এই সিস্টেমে সময়টা নির্দিষ্ট করা লোকাল টাইমে, এবং সময়-এলাকা হল কলকাতা। এবং শুরুতেই দেখুন, স্ট্রিং বা 'Type: string' বলতে বোঝাচ্ছে সিস্টেম এই ফাইল থেকে টেক্সট আকারে ঘোষিত কনফিগারেশন পড়ে, এবং সেই টেক্সট কোনো গাণিতিক মান বা নিউমেরিকাল ভ্যালু নয়, একটা স্ট্রিং বা বর্ণ এবং চিহ্নের সমাহার। এরকম আসতেই পারত 'Type: integer'। শূন্য নম্বর দিনে তথ্যের কাঠামোর আলোচনাটা আর একবার একটু দেখে আসতে পারেন। এবং এই স্ট্রিং-টাকে সিস্টেম তুলে নেয় একটা তথ্য হিশেবে, যেমন আর একটা রকম হল হ্যাঁ-না (yesno), যার বেলায় সিস্টেম কোনো চিহ্ন-সমাহার তোলেনা, শুধু জেনে নেয় একটা বিশেষ প্রশ্নের উত্তর হ্যাঁ না না। আর এক রকম হল তালিকা বা লিস্ট (list)। এই কনফিগারেশন ফাইলগুলোর মধ্যে এক একটা অংশ এক একটা রকমের বা টাইপের থাকতে পারে, সেই অংশটার আগে একটা লাইনে লিখে জানানো থাকে, 'Type: ', এবং তারপর ডানদিকে আসে 'string' বা 'yesno' বা 'list' বা 'integer' ইত্যাদি।

এরকম আর একটা ফাইল, এই '/etc/sysconfig' ডিরেক্টরিতেই, '/etc/sysconfig/keyboard'। এখন কনফিগারেশন ফাইলে আপনার সদ্যলব্ধ চেনাশোনা নিয়ে সেখান থেকে দুচারটে সেটিং একটু বোঝার চেষ্টা করুন। যেমন দেখুন, একটা ভ্যারিয়েবল পাবেন কি-টেবল (KEYTABLE), মানে কিবোর্ড ম্যাপিং। আপনি সুইচগুলো টিপলে যে বর্ণমালা ফুটে ওঠে তার দেশ থেকে দেশে পার্থক্য ঘটে। ধরুন এই কিবোর্ডে খটাখট করে আমরা যখন স্বাভাবিক ল্যাটিন বর্ণমালা পাই, সেখানে একজন ফরাসি ফুটিয়ে তোলে লাকসঁ-তেইগু (é) বা লাকসঁ-সিরকোফ্লেক্স (ê) গোছের অ্যাকসেন্ট চিহ্ন, একজন জর্মন সেখানে ওমলাও (ö) গোছের চিহ্ন সব টাইপ করে, আমাদের সেগুলো আনতে হলে আলাদা করে বাইরে থেকে অ্যাকসেন্টেড ক্যারেকটার ঢোকাতে হয়। এগুলো সব ঘটে কিবোর্ড ম্যাপিং বদলে। এগুলো নিয়ে বেশি ভুলভাল বকলে, সায়মিন্দু আমায় যা-তা করবে, আরো যেদিন এল আমার এখানে, তার আগের দিন মাত্র উনপঞ্চাশখানা ফুচকা খাওয়ার পর ওর পেট একটু বেশি সোচ্চার থাকায় ওকে খাইয়েছিলাম বর্ণহীন স্বাদহীন গন্ধহীন পেঁপে দেওয়া ট্যালট্যালে স্টু, তারপর থেকেই আর কখনো ফোন করেনি আমায়।

আমার মেশিনে যেমন ম্যাপিং-টা হল 'KEYTABLE="us.map.gz"', এর মানে 'gzip' করে কুঁকড়ে ছোট করে রাখা একটা ফাইল, যার নাম 'us.map.gz', সেটাকেই কিবোর্ড-ম্যাপ হিশেবে পড়বে সিস্টেম। এবং সেই ফাইলটা আছে কোথায় সেটাও দেওয়া আছে এই '/etc/sysconfig/keyboard' ফাইলে, জায়গাটা হল '/usr/share/kbd/keymaps' নামের ডিরেক্টরি। এই ডিরেক্টরির মধ্যে আছে ছটা সাবডিরেক্টরি, ছয় ধরনের কম্পিউটার গঠন বা আর্কিটেকচারের জন্যে। তার মধ্যে একটা হল 'i386', ইনটেল ধাঁচের আর্কিটেকচারগুলোর জন্যে। তার মধ্যে অন্তত দুটো নামের সঙ্গে আমাদের পরিচিতি হয়েছে আগেই ভোরাক (dvorak) আর কোয়েরটি (qwerty), শূন্য নম্বর দিনে আমরা নানা ধরনের কিবোর্ড গঠনের কথায় বলেছিলাম। এই '/usr/share/kbd/keymaps/i386/qwerty' ডিরেক্টরির মধ্যে আছে মোট সাতাত্তরটা কিটেবল, প্রত্যেকটাই ওইরকম 'gzip' করা কোঁকড়ানো ফাইল, তার একটা হল 'us.map.gz'। এটাকে বড় মানে স্বাভাবিক করে নিয়ে পড়ুন, 'gunzip us.map.gz' কমান্ড দিয়ে। আপনার কিবোর্ডের কোন বোতামে কোনটা আসবে তার গঠনটা দেওয়া আছে।

আমাদের জিএলটির অয়নের বাড়িতে ম্যানড্রেক ৮.২ ইনস্টল করার সময়ে খুব ঝামেলায় পড়েছিলাম এই কিটেবিলের জন্যে। আমি ইংরিজিতেও ব্রিটিশ বানানের জায়গায় মার্কিন বানান ব্যবহার করি, তাই সরাসরি ইউ-এস সেটিং করি, ওর অভ্যেশ ব্রিটিশ বানানে, তাই গোড়া থেকেই ব্রিটিশ সেটিং করছিলাম। তখন তো বুঝিনি, তারপর কী একটা কনফিগারেশন ফাইল বদলাতে গিয়ে দেখি, উল্টোপাল্টা সব অক্ষর আসছে। ঠিক মনে নেই এখন, বোধহয় '$' টাইপ করতে গেলেই, মানে শিফট টিপে '4' সুইচটা টিপলেই '£' আসছিল, বা এইরকম কী একটা। এই গোটা ডিরেক্টরিটা দেখুন, মেরা নাম জোকারের ডিরেক্টরি একদম, জাপানি জুতো ইংলিস্তানি পাতলুন রুশি টুপি সবই আছে। হিন্দুস্তানি দিল, না-ভাই, সেটার জন্যে আপনাকে কিছু বলার আছে, একদম শেষে গিয়ে বলব। '/etc/sysconfig' ডিরেক্টরির এই 'keyboard' ফাইলে আরো পাবেন কিবোর্ডের রিপিট রেট, ডিলে টাইম ইত্যাদি সবই। নিজে পড়ে দেখুন। রিপিট রেট বা পৌনপুনিকতার হার বলতে বোঝায় কিবোর্ডের একটা চাবি টিপে রেখে দিলে কী হারে সেই বর্ণটা পরপর ফুটে উঠতে থাকবে। আর ডিলে টাইম বা স্থগিত-কাল বলতে চাবিটা টিপে রেখে দেওয়ার কত সময় বাদে ওই পৌনপুনিকতাটা শুরু হবে। এই '/etc/sysconfig/keyboard' ফাইলেই আর একটা জিনিষ পাবেন, কিবোর্ড আইডেন্টিফায়ার, আমার সিস্টেমে সেটা হল '`english-us,pc104`'। আপনি কি এটা দেখে কিছু মনে করতে পারছেন? ইনস্টলেশনের সময়, কী একটা আপনাকে দিতে হয়েছিল না? যদি না-পারেন, চেষ্টা করুন, করতেই থাকুন। এই বইটায় আমি ঠিক যেটা চেয়েছি, এই বইটার বাইরে বেরিয়ে যেতে পারা, আপনার নিজের মাথার মধ্যে যাত্রাপথে। কদ্দিন হয়ে গেল বলুন তো, সেই শুটকো মোদোমাতাল কাঁচাপাকা দাড়ির দোমড়ানো আধবুড়োটা বলেছিল, 'ভাবো ভাবো ভাবা প্র্যাকটিস করো'। এইরকম '/etc/sysconfig/mouse' ফাইল জুড়ে সিস্টেম আপনাকে শোনাবে ইঁদুরিনীর ঠিকুজি — দ্যু উ ত্রোয়া শোজ কে জ সে দে সুরিস — ইঁদুর বিষয়ে দুটি বা তিনটি কথা যা আমি জানি (কার্টসি গদার)। দুটি বা তিনটির চেয়ে আদতে অনেক বেশি কথাই শোনাবে আপনাকে, পড়ে দেখুন। যাকগে, এই '/etc/sysconfig' অনেক হল, এবার যাওয়া যাক সরাসরি '/etc' ডিরেক্টরির কিছু কনফিগারেশন ফাইলের কথায়।

২.৩।। '/etc' ডিরেক্টরির কনফিগারেশন ফাইলেরা

এবারেও আমরা কাজে লাগাব 'writefiles' নামে আমাদের বানানো ব্যাশস্ক্রিপ্ট। '/etc' ডিরেক্টরির উপর এটা চালিয়ে পাওয়া 'dir.etc.textfile' ফাইলটার সাইজ হয়েছে ২.৮ এমবি, এবং এর প্রথম এন্ট্রি '`1. DIR_COLORS`' আর শেষ এন্ট্রি '`227. zshrc`'। তার মানে দুশো সাতাশখানা কনফিগারেশন ফাইল আছে। তার প্রথমটা 'DIR_COLORS', স্থির করে দেয়, 'ls' মেরে আমরা যখন ফাইল ও ডিরেক্টরির তালিকা দেখি তখন তিনি কোন ধরনের ফাইলকে কী রঙে রাঙাবেন। রঙবাজির গোটা ছকটা একবার দেখে নিন ফাইলটা থেকে। আর শেষতম মানে দুশো সাতাশতম ফাইল হল 'zshrc', নামের মিল থেকে কিছু আন্দাজ করতে পারছেন, ঠিক 'bashrc' ফাইল যেমন ব্যাশের ছকটা রাখে, সেইরকম 'zshrc' করে জেড-শেলের, আপনি চাইলে ব্যাশের বদলে যা ব্যবহার করতে পারেন। ডিরেক্টরিগুলোকে ও নিজেই বাদ দিয়ে দেয়, স্ক্রিপ্টটা চালিয়ে দেখুন। একটা করে এরর মেসেজ পাঠায় স্ক্রিনে, ডিরেক্টরি ফাইল ক্যাট করতে পারছো, এমনকি রুট সেটা চাইলেও। মনে আছে, আমরা বলেছিলাম, ডিরেক্টরি ফাইল দেখার অধিকার কারনেল ছাড়া আর কারুর নেই।

২.৩.১।। হোস্ট কনফিগারেশন

'/etc/host.conf' বা '/etc/hosts' বা '/etc/hosts.allow' বা '/etc/hosts.deny' বা এই ধরনের আরো কিছু ফাইল — এই ফাইলগুলো কী করে সে ব্যাপারে নামগুলো দেখেই একটা মাইল্ড আন্দাজ করতে পারছেন। একটু আগে যে হোস্টনেম মানে মেশিনের আত্মপরিচিতির কথা বলছিলাম, সেই ব্যাপারটা। এটা আরো দরকার পড়ে যখন আপনার মেশিন একটা নেটওয়ার্কে যুক্ত থাকে। আজকের ১.১ নম্বর সেকশনে আমার মেশিনের নামের যে কথাটা বলছিলাম, এখানে দুটো লাইন তুলে দিই আমার সুজে সিস্টেমের '/etc/hosts' ফাইল থেকে, মিলিয়ে নিন।

```
# IP-Address  Full-Qualified-Hostname  Short-Hostname
127.0.0.1       mahammad.local mahammad
```

এই ইন্টারনেট প্রোটোকল (IP) এবং ট্রান্সমিশন কন্ট্রোল প্রোটোকল (TCP) আপনি একদম গুলে খেয়ে ফেলতে পারবেন জাস্ট খেজুরগাছে হাড়ি বেঁধেই। আমাদের তথাগত তো তিনদিনে শিখল কী করে একটা সাইবার কাফের

গোটা নয়েক মেশিন নিজেদের মধ্যে ল্যান-জালে আর নেটের সঙ্গে জগত-জোড়া-জালে (ওয়ার্ল্ড-ওয়াইড-ওয়েব, নামটা হেভি নামল না? 'www'-র জায়গায় 'জজজ'?) যুক্ত করে ফেলার কায়দা, সেরেফ এই ম্যানপেজ পড়ে। দেখুন, কাঠামোটা ও আপনাকে নিজেই বলে দিচ্ছে, প্রথমে আইপি ঠিকানাটা, তারপরে গোটা হোস্টনাম, তারপরে তার ছোট সংস্করণ। এই প্রথমটা মানে আইপি-ঠিকানাটা হয় চার-বাইটের এবং কাঠামোটা হয় ক.খ.গ.ঘ আকারের। ক বা খ বা গ বা ঘ হয় ০ থেকে ২৫৫ অব্দি কোনো একটা সংখ্যা। কেন হয় এমন, দেব আপললন, কেন এমন হয়? — শম্ভু মিত্রের বটতলা প্যারডি নিজেই নিজেকে একবার শুনিয়ে এই গোটা লেখাটা শূন্য থেকে শুরু করুন, যদি মনে করতে না-পারেন। টিসিপির নাম থেকে আন্দাজ করতে পারছেন জজজ বা ল্যানজাল বেয়ে একটা মেশিন থেকে আর একটা মেশিনে তথ্য দৌড়োদৌড়ির প্রক্রিয়াটাকে নিয়ন্ত্রণ করার একটা প্রোটোকল। এই তথ্যটা পাঠায় কোনো একটা অ্যাপ্লিকেশন বা প্রয়োগ সফটওয়ার, কোনো একটা ব্রাউজার কিম্বা একটা ইমেল-প্যাকেজ কিম্বা এই ধরনের কিছু। সেখানেও ভাবুন, আপনি একটা নাম দেন আপনার ব্রাউজারের কাছে বা ইমেল ঠিকানা লেখেন, ধরুন আপনি খুঁজছেন গ্নু-র ওয়েবসাইট 'www.gnu.org', এইটা কিন্তু আপনার বোঝার আর মনে-রাখার সুবিধের জন্যে বানানো একটা নাম, আপনি এই নামটা দিয়ে এন্টার মারার পরে, বা ইঁদুর দিয়ে হলে কোনো একটা লিংকে ক্লিক করার পরে, এই নামটা চলে যাচ্ছে কোনো একটা নেম সার্ভারের কাছে, সেই গাবদা মেশিনগুলো, যাদের কাজই এই নামগুলো থেকে মূল আইপি ঠিকানাটা বার করা এবং সেই ঠিকানার মেশিনের ঠিকানার ফাইলটা পাওয়ার জন্যে সবচেয়ে সহজ পথটা খুঁজে বার করা। এই মূল ঠিকানাটা জানার জন্যে পিং করতে পারেন। ধরুন নেটে কানেক্ট করেছেন, সে 'wvdial' বা 'kppp' বা 'kinternet' ইত্যাদি যা দিয়েই হোক, কানেকশনটা তৈরি হয়েছে, এই অবস্থায় আপনি পিং কমান্ড দিয়ে মূল আইপি অ্যাড্রেসটা জানতে পারেন। যেমন এইমাত্র '`ping www.gnu.org`' দিয়ে দেখলাম, ও দেখাল '`199.232.41.10`'। এটা একটা বাস্তব মেশিনের বাস্তব ফাইলের নেটওয়ার্ক ঠিকানা। ধরুন এবার আপনি ভাবলেন, গ্নু.অর্গের ওই ব্যাঁকাশিঙ ভ্যাবলা গ্নু-কে একটা মেল করবেন, বেচারা ওয়েট করে থাকে আপনার এলাকার মুক্ত সফটওয়ার আন্দোলনের খবর পাওয়ার জন্যে, ওর মেল আইডি তো জানেনেই — 'gnu@gnu.org'। এবার একটা দীর্ঘ চিঠি লিখলেন ফ্লস (FLOSS — **F**ree/**L**ibre-**O**pen-**S**ource-**S**oftware) নিয়ে, শেষে চিঠিটা পাঠাতে বললেন আপনার মেল সফটওয়ারকে। সে গ্নু.অর্গের আইপি ঠিকানায় গ্নু নামক ব্যবহারকারীর মেইলবক্সে ফাইলটা পাঠিয়ে দিল। ব্রাউজার ফাইল তথ্য নামাতে চাওয়া বা মেল প্যাকেজ পাঠাতে চাওয়া মাত্রই টিসিপি তথ্যটা পাঠিয়ে দিল আইপি-র কাছে, এবার আইপি দেখবে একটা মেশিন থেকে আর একটা মেশিনে তথ্য ঠিকঠাক যাচ্ছে কিনা, বা একটা থেকে পাঠানো তথ্য অন্য মেশিন বুঝতে পারছে কিনা, ইত্যাদি।

এবার ফেরত আসা যাক আমাদের '/etc/hosts' ফাইল থেকে তোলা ওই দুই লাইনে, দেখুন এই গোটা কাঠামোটা, আপনার মেশিনের নিজের ডোমেনে অবস্থান সহ আপনাকে জানিয়ে দিচ্ছে, আমার মেশিনের নাম যেমন '`mahammad`' আগেই বলেছি, আর ডোমেন তো লোকালডোমেন। এবার প্রশ্ন হল, এখানে যতটুকু আলোচনা আসছে তার বেশি আপনি বুঝবেন বা এগোবেন কী করে? এবার কয়েকটা লাইন তোলা যাক '/etc/host.conf' ফাইল থেকে। বরং ফাইলটা বেশ ছোটই, গোটাটাই তুলে দেওয়া যাক, মধ্যে পড়ার সুবিধের জন্যে ফাঁকা লাইনগুলো বাদ দিয়ে।

```
# /etc/host.conf - resolver configuration file
# Please read the manual page host.conf(5) for more information.
# The following option is only used by binaries linked against
# libc4 or libc5. This line should be in sync with the "hosts"
# option in /etc/nsswitch.conf.
order hosts, bind
# The following options are used by the resolver library:
multi on
```

এটায় দেখুন, দু নম্বর লাইনটা — সেই খেজুরগাছের বৃত্তান্ত। দিয়েছে দেখুন, 'host.conf(5)'। তার মানে ম্যানপেজের পাঁচ নম্বর সেকশনে থাকবে এই ম্যানুয়ালটা। কেন পাঁচে? কেন সাতে বা চারে বা উনচল্লিশে নয়? এই ম্যানপেজগুলো পড়ার একটা ঝামেলা হল, সবসময়েই ম্যান প্যাকেজ মানে '`man`' দিয়ে পড়তে হয়। একটা ম্যানপেজ আবার আর একটা পড়তে বলে। আর সেটাও আবার ম্যান করেই পেতে হয়। দু-একটা ম্যানপাতা আপনি

নয় ওই ‘rman’ করে বানিয়ে নিলেন, কিন্তু কত বানাবেন? ম্যানপেজের পরিমাণ তো জানেন। আমার মেশিনের সুজে সিস্টেমে আছে মোট ম্যানপেজ আছে সাড়ে ছ হাজার। তিন নম্বর সেকশনে সবচেয়ে বেশি, ৩১৮৮-খানা, আর সবচেয়ে ছোট ছয় নম্বর সেকশনটা, ৪৮-খানা। ছয় নম্বর দিন থেকে দেখে নিন, কোন সেকশনে কী আছে। তার চেয়ে সবগুলোকে একসঙ্গে যদি আপনি ওয়েবপেজ বা এইচটিএমএল (*.html) ফাইল করে নেন, তাতে আপনি ব্রাউজ করতে পারবেন আপনার ব্রাউজার দিয়েই। আপনার ‘/usr/share/man’ ডিরেক্টরির ভিতর ‘man1’, ‘man2’ ইত্যাদি সবকটা সাবডিরেক্টরির সমস্ত ম্যানপেজগুলোই দেখুন ‘gzip’ করা কোঁকড়ানো ফাইল। সব কটা সাবডিরেক্টরিকে কপি করে আনুন একটা কোনো ডিরেক্টরিতে, ধরুন ‘browse.manpages’ নামে। এবার প্রতিটি সাবডিরেক্টরিতে একবার করে ঢুকে নিচের এই ব্যাশস্ক্রিপ্টটা চালান, এক এক বারে আপনার একটা গোটা সেকশনের সবকটা ম্যানপেজ ওয়েবপেজ ফরম্যাটে চলে আসবে। নটা ডিরেক্টরিতে নবার চালিয়ে গোটা কাজটা হয়ে যাবে, সাড়ে ছ-হাজার ফাইলের ‘gz’ থেকে ভেঙে বার করা, তারপর তাকে ওয়েবপেজ করা, তারপর মূল ফাইলগুলো উড়িয়ে দেওয়া। এটা আমার সিস্টেমে ‘changeman’ নামে একটা স্ক্রিপ্ট বানিয়ে, তাকে ‘`chmod +x`’ করে এক্সিকিউটেবল বানিয়ে আমার হোমের বাইনারি ডিরেক্টরিতে মানে ‘/home/dd/bin’-এ রেখে দেওয়া আছে। ‘/home/dd/bin’ ডিরেক্টরি ‘$PATH’-এ থাকায় শেল সবসময়েই এটা খুঁজে পাবে।

```
#!/bin/bash
for i in `ls *.gz`
    do gunzip $i
done
for j in `ls *`
    do rman $j -f html > $j.html
    rm -f $j
done
```

খতিয়ে দেখুন তো, প্রায় গোটাটাই আন্দাজ করতে পারেন কিনা। না-হলেও চিন্তা নেই, একটু বাদেই পারবেন, তবে, যা মনে হচ্ছে, আর একটা দিন বোধহয় যোগ করতেই হবে, শেল স্ক্রিপ্টের জন্যে। আগের ‘writefiles’ শেলস্ক্রিপ্টের থেকে এখানে কেবল একটা নতুন কনসেপ্ট বা ধারনাই ব্যবহার করা হয়েছে। সেইটুকু বলে দিই। সেটা হল ওই দ্বিতীয় লাইনে, ‘`for i in `ls *.gz``’, কোটচিহ্ন বা উর্দ্ধকমাটা খেয়াল করুন — ওটা কিন্তু আমাদের অভ্যস্ত ফ্রন্টকোট বা প্রগতিশীল সামনে-কমা নয়, প্রতিক্রিয়াশীল পিছনে-কমা বা ব্যাককোট। সামনে-কমাটা পান কোয়েরটি কিবোর্ডে ডান হাতের কড়ে আঙুল ডানে সরিয়ে, ‘L’-কে এক নম্বর ধরে ডানে তিন নম্বর চাবিতে। আর এই পিছনে-কমাটা পাওয়া যায় একদম বাঁদিকে উপরে, ঠিক ‘Esc’ চাবির নিচে, দেখুন দুটো চিহ্ন দেওয়া আছে চাবিটায়, উপরে ‘~’ আর নিচে ‘`’। মানে শিফট চেপে এই চাবিটা টিপলে আসে ‘~’, আর এমনিতে আসে ‘`’। সংশোধনবাদী বিচ্যুতিতে ভোগা এই পশ্চাৎপ্রবণ কমার জীবনে মরণে একটাই চরিতার্থতা, আদেশ অনুবাদ, কমান্ড সাবস্টিটিউশন। এখানে এই ‘``ls *.gz``’ অংশটার মানে, কমান্ড প্রম্পটে নিছক ‘`ls *.gz`’ আদেশটুকু দিয়ে এন্টার মারলে যা ফুটে উঠত স্ক্রিনে সেটাকেই এখানে তৈরি করে দিচ্ছে এই প্রতিক্রিয়াশীল কমাদ্বয়। কী পেতাম আমরা? এই কমান্ড দিলে? আমাদের সামনে স্ক্রিনে ফুটিয়ে তুলত এই ডিরেক্টরিতে সমস্ত ‘`gz`’ এক্সটেনশন সম্পন্ন ফাইলের তালিকা। তার থেকে আমরা আমাদের ভ্যারিয়েবল ‘`i`’-কে পড়ছি মানে — আমরা শুধু জিজিপকৃত ফাইলগুলোকেই তুলছি একের পর এক। ওই একই কেতা ব্যবহৃত হয়েছে দেখুন, আবার পাঁচ নম্বর লাইনে। এটা সরাসরিও করা যেত, কিন্তু আমার মধ্যে ওই গোপন ধান্দাবাজিটা তো কাজ করছেই, একটু পরেই যা আসবে তার সঙ্গে আপনাদের পরিচিত করে রাখতে চাইছি।

এবার কী মজা দেখুন সবগুলো ম্যানপেজই ‘*.html’ ফাইল হয়ে গেছে, টেক্সট মোডে ‘links’ বা ‘lynx’ দিয়ে, কিম্বা গুই হলে ‘konqueror’ বা ‘galeon’ বা ‘mozilla’ বা যা আপনার খুশি তাই দিয়ে ব্রাউজ করুন। ‘links’ বা ‘lynx’ দিয়ে চমৎকার ফাইলসিস্টেমও ব্রাউজ করা যায়। ধরুন আপনি দিলেন ‘`lynx /`’ এবং এন্টার মারলেন, দেখবেন আপনার রুট ডিরেক্টরির সমস্ত ডিরেক্টরি এবং ফাইল পরপর ফুটে উঠেছে, এবার ডাউন অ্যারো বা নিম্নমুখী তীরটা (↓) টিপলে দেখবেন একটার পর একটা ডিরেক্টরি হাইলাইটেড হয়ে উঠছে, এবং সেখানে রাইট বা ডানপন্থী অ্যারো (→) টিপলে দেখবেন সেই ডিরেক্টরির মধ্যে ঢুকে গেলেন, সেখানে আবার একই ভাবে সব দেখাচ্ছে। যাকগে, এসব

তো আপনি 'man lynx' পড়ে জেনেই যেতে পারবেন। যা বলছিলাম, এবার আপনার এই সাড়ে ছ-হাজার ম্যানপেজ আপনি ইচ্ছেমত ব্রাউজ করতে পারবেন, উইন্ডোশপিং করতে পারবেন মানে দরকার পড়লে কোনো জায়গায় উইনডোজ থেকেও পড়তে পারবেন, তখন ব্রাউজারটা হবে স্বাভাবিক ভাবেই আইএক্সপ্লোরার বা নেটস্কেপ দিয়ে, ইচ্ছে হলে খুব সহজে প্রিন্ট আউট নিতে পারবেন। বিশুদ্ধ জ্ঞানপিপাসা — তাও চলবে, পড়ুন না, কত আপনার দম, পড়ে যান। এই ওয়েবপেজ আকারে গোটা ম্যানপেজ ডিরেক্টরিটার আকার দাঁড়াল পঁয়ত্রিশ এমবির মত। কোনো কোনো ডিস্ট্রো এইরকম ব্যবস্থাটা এমনিতেই করা থাকে, নিজে আর করে নিতে হয়না।

এই ম্যানপেজ ডিরেক্টরিতে গিয়ে সমস্ত ফাইল ক্যাট করে দিলাম, দিয়ে সেটাকে পাইপ করলাম 'wc'-র কাছে, মানে কমান্ডটা হল, 'cat *.html | wc'। এতে মোট লাইন শব্দ আর চিহ্নের সংখ্যা পেয়ে গেলাম। সমস্ত সেকশনের ম্যানপেজ মিলিয়ে মোট শব্দের সংখ্যা দাঁড়াল তেত্রিশ লাখ, মানে তিনশো করে শব্দ পাতাপিছু ধরলে এগারো হাজার পাতার মত। এই একই ভাবে মোট হাউটুর মোট সাইজ পেলাম সাতান্ন লাখের মত, মানে ধরুন উনিশ হাজার পাতা। এই তিরিশ হাজার পাতা টেক্সট, আপনি রোজ একশো পাতা গ্নু-লিনাক্স কমিউনিটির সবাই মিলে বানিয়ে রেখেছে ফর ইয়োর আইজ ওনলি। আপনি এবার ইচ্ছে হলে এই ম্যানুয়ালের ওয়েবপেজ ডিরেক্টরিতে একটা ইনডেক্স ওয়েবপেজও বানিয়ে নিতে পারেন, সরল সাদামাটা ওয়েবপেজের এইচটিএমএল ফর্ম্যাটিং শেখাটা কোনো ব্যাপার না। দু-তিনটে জটিলতাহীন ওয়েবপেজ বেছে নিন, এবং ইম্যাক্সে তাদের খুলুন একটা কনসোলে। এবার একটা করে ট্যাগ বদলান, আর অন্য কনসোলে একটা ব্রাউজার দিয়ে দেখুন সেটায় কী বদল ঘটছে ফাইনাল ওয়েবপেজে। আমি আমার হাটুরে অর্ধশিক্ষিত রকমে এভাবেই ওয়েবপেজ লেখা শিখেছিলাম, ঠেকে শিখতে হয়েছিল। দারিদ্র জিনিষটা এইসব ব্যাপারে বেশ কাজের। আমাদের বই মার্জিন অফ মার্জিনের সাইটের ওয়েবপেজ লিখতে হবে, এদিকে কাউকে দিয়ে লেখানোর পয়সা নেই, বই বার করতে গিয়েই প্রচুর ধার হয়ে গেছে, আর খুব দ্রুত সেটা করতে হবে — আর কী, জয় শ্রীরাম বলে লেগে গেলাম — না না, এখন তো টি-টোটালার গত সাত বছর ধরে, রাম আমি তখনো খেতাম না। ওয়েবপেজ শেখার এটা জংলি পদ্ধতি খুবই, কিন্তু বেশ কাজে দেয়।

আমাদের '/etc/host.conf' ফাইল যেমন বলেছিলি, সেই 'host.conf' ফাইলের ম্যানপেজ, যেটা রাখা আছে কনফিগারেশন ফাইলের ম্যানপেজের সেকশনে, মানে পাঁচ নম্বরে, আমাদের 'changeman' শেলস্ক্রিপ্ট দিয়ে এইমাত্র বানানো 'host.conf.5.html' ফাইলটা থেকে দুটো লাইন তুলি, "The file */etc/host.conf* contains configuration information specific to the resolver library. It should contain one configuration keyword per line, followed by appropriate configuration information. The keywords recognized are *order*, *trim*, *multi*, *nospoof*, and *reorder*."। এই লাইন তিনটে, এবং তার পরেও যা আছে, কিছুটা চিবিয়ে দেখলাম, মাইরি বলছি, আমি প্রায় কিছুই বুঝছি-না, আপনি যদি বুঝতে চান, নেটওয়ার্কিং সংক্রান্ত হাউ-টুগুলো পড়ে ফেলুন। আর যদি বিপুল ফান্ডা লড়িয়ে ফেলতে পারেন, এবং কাছাকাছি থাকেন, তাহলে জানাবেন, প্লিজ। আমার কিছুদিনের মধ্যেই একটু নেটওয়ার্কিং-সাক্ষর হওয়ার ইচ্ছে আছে। এই '/etc/host.conf' ফাইলটা আসলে নেটওয়ার্ক ডোমেইন সার্ভারকে বলে দেয়, কোথায় কোথায় কী ক্রম অনুযায়ী হোস্টনামের তালিকা খুঁজতে হবে। সচরাচর এই ক্রমটা হল প্রথমে '/etc/hosts', তারপরে নেম সার্ভার। এইটুকু থেকেই নিশ্চয় বুঝতে পারছেন, আমাদের মত একা একলা মেশিনের সিস্টেমের লোকেদের জন্যে এটা অনেকটা কলকাতার বাঙালির কোট বা গলার জাঙ্গিয়া মানে টাইয়ের মত। আমাদের ধরনের মেশিনে নেটওয়ার্কিং সংক্রান্ত যে ফাইলটা মাঝে সাঝে কাজে লাগে সেটা হল '/etc/resolv.conf'। এই ফাইলটা কারনেলকে বলে দেয় কোনো একটা প্রোগ্রাম একটা নির্দিষ্ট আইপি ঠিকানা পেতে চাইলে সেটার জন্যে কোন নেম সার্ভারকে জিগেশ করতে হবে। আমি প্রথম যখন ম্যানড্রেক ছেড়ে সুজেতে এলাম, 'kppp' চালানো যাচ্ছিল না। আর নেটে কানেক্ট করার আর কোনো উপায় তখন জানতাম না। যখনি 'kppp' চালাতে যাই, সিস্টেম বলে, হয় '/etc/resolv.conf' পাওয়া যাচ্ছেনা, নয়তো সেই ফাইলের মালিকানা ও অনুমতিতে কোনো গন্ডগোল আছে। আমি রুট হয়ে 'touch' করে ফাইল বানিয়ে দিচ্ছিলাম, এবং 'chmod' করে সেটাকে সর্বসাধারণের ব্যবহারযোগ্য করে নিচ্ছিলাম, প্রতিবার 'kppp' চালানোর সময়। এতে কাজ চলে যাচ্ছিল। কিন্তু কী ফ্যাসাদ ভাবুন, শেষে একটা স্ক্রিপ্ট বানিয়ে নিয়েছিলাম, যাতে বুট করার সময়েই এটা হয়ে যায়। আর যদ্দুর মনে পড়ছে '/etc/rc.local' ফাইলে স্ক্রিপ্টের নামটা লাগিয়ে দিয়েছিলাম, এখানে যে স্ক্রিপ্টের নাম থাকে সেটাকে সিস্টেম বুট করার সময় চালিয়ে নেয়, আগেই

বলেছি। এবং সেই স্ক্রিপ্টটাকে ‘chmod 4755’ করে দিয়েছিলাম, একে বলে রুট আইডি সেট করা, যাতে সিস্টেম মনে করে যে প্রোগ্রামটা রুটই চালাচ্ছে, আদতে সেটা যে-ই চালাক না কেন। এখন তো আর ‘kppp’ ব্যবহারই করিনা প্রায়। কিন্তু সমস্যাটা কেন হয়েছিল আমি আজো বুঝিনি। যতদূর মনে হয় সিস্টেম সিকিউরিটি সংক্রান্ত কোনো একটা খ্যাঁচ আছে সুজেতে, যেমন আছে ‘locate’ ডেটাবেস ব্যবহার করায়। নিজের থেকে চালিয়ে নিতে হয়, আলাদা করে। সিস্টেম নিজে থেকে লোকেট প্যাকেজটা চালু করেনা।

যাকগে, যে কথা হচ্ছিল, ‘/etc’ ডিরেক্টরির হোস্ট সংক্রান্ত ফাইলেরা। ‘/etc/hosts’ ফাইলে থাকে লোকাল নেটওয়ার্কের, মানে এই সিস্টেমটা যে নেটওয়ার্কের মধ্যে রয়েছে, সেখানকার মোট হোস্টের নাম এবং ধাম। আমার মেশিন যেহেতু একা একলা, স্ট্যান্ড-অ্যালোন, পিসি, তাই সেখানে ওই তালিকায় ছিল একটা মেশিনেরই আইপি ঠিকানা ‘127.0.0.1’, যার আমার দেওয়া মানববোধ্য নাম ‘mahammad.local’। আর ‘/etc/hosts.allow’ এবং ‘/etc/hosts.deny’ ফাইলে কী খায় এবং কী কী মাথায় দেয়, গলাতেও দিতে পারে, রজ্জু জাতীয় কিছু, সেটা প্লিজ নিজে ম্যানপেজ পড়ে দেখে নিন — অত অত ম্যান ওয়েবপেজ কাঁদবে তো?

২.৩.২।। ‘/etc/fstab’ এবং ‘/etc/mtab’ তথা ফাইলসিস্টেমের কনফিগারেশন

এদুটো নিয়ে মুরগিতে কোনো ফ্লু হওয়ার চান্স নেই, এইসব ডামাডোল শুরু হওয়ার আগেই ছয় সাত আট নম্বর দিনে আমরা কুপিয়ে রান্না করে রেখেছি। ফাইলসিস্টেমের আর পার্টিশনের আলোচনায়। ‘/proc’ ডিরেক্টরির কথা তো আমরা আগেই বলেছি, আট নম্বর দিনে। কারনেলের একটা সমাজসেবা মূলক কাজ হিশেবে ভাবতে পারেন ‘/proc’ ডিরেক্টরিটাকে। সমাজ মানে আমরা। আমরা যাতে গোটা সিস্টেম জুড়ে চলতে থাকা অজস্র বহুমুখী প্রক্রিয়ার মোট বাস্তবতাটা কোনো এক ভাবে ধরে উঠতে বুঝে উঠতে পারি সেইজন্যে কারনেলের বানিয়ে দেওয়া একটা ইন্টারফেস, সিস্টেমের সঙ্গে মুখোমুখি হওয়ার একটা রিসেপশন ডেস্ক। খটাখট করে ইন্টারকম টিপবে আর বলে দেবে কোন বাবু আর কোন বিবি কী করছেন, কোথায় আছেন, কী তার অবস্থা এখন। বলে দেবে বিভিন্ন সিস্টেম নিয়ন্তা বা প্যারামিটারগুলোর এই মুহূর্তে কী আকার। এই ‘/proc’ ডিরেক্টরিটা একটা ডামি বা নকল বা ছদ্ম ফাইলসিস্টেমের মত। চার নম্বর দিনে আমরা যেমন ক্লদ শ্যাননের ব্যবহার করা বিদ্যুৎব্যবস্থার অ্যানালগের কথা বলেছিলাম, যেখান থেকে অ্যানালগ কম্পিউটারের ধারণাটা এসেছিল, সেইরকম, মূল বাস্তব জ্যান্ত সিস্টেমের একটা অ্যানালগ হিশেবে ভাবতে পারেন ডিরেক্টরিটাকে।

বাজার অর্থনীতির জারগনে ফ্লো বা প্রবাহ ভ্যারিয়েবল আর স্টক বা জমা ভ্যারিয়েবলের একটা ধারণা আছে। ধরুন একটা বাসস্টপ। সেখানে গোটা দিনটা, ভোর থেকে রাত্তির সবসময়েই লোক দাঁড়িয়ে থাকছে। কিন্তু কোনো একটা লোকই সারাদিন ধরে বাসস্টপে দাঁড়িয়ে থাকছে না। এই সারাদিন যাত্রীর ভিড়টা একটা প্রবাহ বা ফ্লো। আর একটা বিশেষ মুহূর্তে বাসস্টপে দাঁড়িয়ে থাকা লোকের সমাহারটা একটা জমা বা স্টক। কিন্তু জ্যান্ত ফ্লো-টাকে সময় থেকে সময়ে স্টক মেপেই কিন্তু অনেকটা বুঝে নেওয়া যাবে। যেমন, যে কোনো একটা মুহূর্তে, সল্টলেক এফডি স্টপে বা শোভাবাজার মেট্রো স্টপে, দুটো স্টক নাটকীয় ভাবে আলাদা হবে, ঠিক তাদের নিজস্ব ফ্লো-পার্থক্যের সঙ্গে সঙ্গতি রেখে। ঠিক সেইরকম, এই ‘/proc’ ডিরেক্টরিটাকে একটা গ্নু-লিনাক্স সিস্টেমের জ্যান্ত প্রক্রিয়াগুলোর সমাহারের ফ্লো-টাকে কিছু নির্দিষ্ট নিরিখ বা নিয়ন্তা বা প্যারামিটারের স্টক দিয়েই পরিমাপ করে। যে কোনো মুহূর্তে সেই পরিমাপগুলো দিয়ে কী ভাবে আমরা একটা সিস্টেমকে বোঝার চেষ্টা করি — সে কথা তো আগেই বলেছি। শুধু আমরা বুঝছি তাই নয়, সিস্টেমের কাজকর্ম সংক্রান্ত অনেকগুলো ইউটিলিটি বা উপযোগিতা প্যাকেজই কাজ করে ‘/proc’ ডিরেক্টরির এই স্টক ভ্যারিয়েবলগুলোর মানগুলো নিয়ে।

‘/etc/mtab’ ফাইল হল ঠিক ওই ‘/proc’ ডিরেক্টরির একটা শিশু-সংস্করণ। গোটা সিস্টেম এর কাজ শুধু পার্টিশন আর তাদের মাউন্ট প্রক্রিয়াকে খেয়াল রাখা। একটা বিশেষ মুহূর্তে কোন কোন পার্টিশনের কোন কোন ফাইলব্যবস্থা মাউন্ট করা আছে সম্পূর্ণ ঐক্যবদ্ধ ফাইলব্যবস্থার মধ্যে, সেটার হিশেব রাখাই কাজ এই ‘/etc/mtab’ ফাইলের। এই ফাইলটা শুধুই বদলাতে থাকে ‘/proc/mounts’ ফাইলের সঙ্গে তাল রেখে, কোনো কোনো ডিস্ট্রোতে এই ফাইলটার নাম ‘/proc/mount’-ও হয়। আসলে এই ‘/etc/mtab’ ফাইলটা ওই ‘/proc/mounts’ ফাইলেরই প্রতিরূপ ধরা যায়। আর এই দুটো ফাইল বদলানোর সঙ্গে সঙ্গে আর একটা ফাইলও বদলাতে পারে, কোন কোন পার্টিশন মাউন্ট হচ্ছে তাতে কী কী ধরনের ফাইল তথা তথ্যব্যবস্থা থাকছে, মানে ফাইলসিস্টেম শব্দটার দু-নম্বর অর্থে, তার সঙ্গে মিলিয়ে, সেই

ফাইলটার কথা আমরা আগেও বলেছি। মনে করতে পারছেন? তার নাম '/proc/filesystems'। আর পার্টিশন মাউন্ট এবং এই তিনটে ফাইলের বদল — এই গোটাটাই ঘটে যে ভিত্তিটার উপর দাঁড়িয়ে সে হল আমাদের পুরোনো চেনা মক্কেল মানে '/etc/fstab'। এই '/etc/fstab' ফাইলে থাকে একটা মেশিনে কোন কোন পার্টিশন মাউন্টযোগ্য তার পূর্বঘোষিত তালিকা। কম্পিউটারে প্রাণপ্রতিষ্ঠা মানে বুট করার সময় সিস্টেম '`mount -a`' কমান্ডটা চালায়। এই অপশন '`-a`' অংশটা সিস্টেমকে বলে দেয় '/etc/fstab' ফাইলে সেই সমস্ত পার্টিশনে মাউন্টের ব্যবস্থা করতে যাদের '`dump`' স্তম্ভে মানে ঠিক শেষের আগের স্তম্ভে এক আছে। এখানে আর একবার আট নম্বর দিনের ৪ নম্বর সেকশনে তুলে দেওয়া '/etc/fstab' ফাইলটার একটা অংশ তুলে দিই, তাহলে আপনাদের মেলাতে সুবিধে হবে বদলগুলোকে।

```
#device    mountpoint        fs        options                dump  fsck
/dev/hdb3  /                 xfs       defaults               1     1
/dev/hdb1  /boot             xfs       defaults               1     2
/dev/hdb2  swap              swap      pri=42                 0     0
/dev/hda1  /mnt/windows/c    vfat      rw,noauto,noexec       0     3
/dev/hda5  /mnt/windows/d    vfat      rw,noauto,noexec       0     3
/dev/hda6  /mnt/slackware    reiserfs  rw,noauto,user,exec    0     3
/dev/hdb5  /mnt/arkive       reiserfs  rw,noauto,exec         0     3
devpts     /dev/pts          devpts    mode=0620,gid=5        0     0
proc       /proc             proc      defaults               0     0
usbdevfs   /proc/bus/usb     usbdevfs  noauto                 0     0
```

এই '/etc/fstab' ফাইলটা একটু ছোট করা, বাস্তব হার্ডডিস্কের সাতটা ভৌত পার্টিশন বাদে সিস্টেমের প্রক্রিয়াগত ভার্চুয়াল পার্টিশন তিনটে 'devpts', 'proc', আর 'usbdevfs' রয়েছে, কিন্তু ফ্লপি আর সিডি-ড্রাইভদুটোর মোট তিন লাইন বাদ দিয়ে দিয়েছি। একবার মিলিয়ে নিন আগের সাত এবং আট নম্বর দিনে আমার মেশিনের সুজে সিস্টেমের পার্টিশন-ব্যবস্থা নিয়ে যা যা বলেছি তার সঙ্গে। '/' আর '/boot' ডিরেক্টরি দুটো মাউন্ট হয় '/dev/hdb3' আর '/dev/hdb1' পার্টিশনদুটোয়, এই দুটোয় আছে এক্সএফএস ফাইলব্যবস্থা। এরপর '/dev/hdb2' হল সোয়াপ পার্টিশন। এবং '/dev/hda1' আর '/dev/hda5' হল দুটো উইনডোজ পার্টিশন, যারা মাউন্ট হয় যথাক্রমে '/mnt/windows/c' আর '/mnt/windows/d' ডিরেক্টরিতে। উইনডোজের দুটো ছাড়াও আরো দুটো পার্টিশন আছে যাদের প্রয়োজন মোতাবেক আমি মাউন্ট করে নিই, তারা হল '/dev/hdb5' এবং '/dev/hda6'। এরা মাউন্ট হয় যথাক্রমে '/mnt/arkive' এবং '/mnt/slackware' ডিরেক্টরিতে।

এবার দেখুন, আগেই বলেছি, আমার সিস্টেমে সুপারমাউন্ট অফ করা আছে, সিস্টেম বুটের জন্যে দরকারিটুকু বাদ দিয়ে আর কিছুই নিজে থেকে মাউন্ট করা হয়না। এই ব্যবস্থা নিজে হাতে 'fstab' লিখে করে নেওয়া যায়, আবার সরাসরি '`supermount -i disable`' কমান্ড দিয়েও করা যায়, সিস্টেমই এফস্ট্যাব ফাইল বদলে লিখে নেয়। লিখে নেওয়াটা ঘটায় ওই '`-i`' অপশনটা। তবে এটা ম্যানড্রেকে আছে মনে পড়ছে, আর কোনটায় আছে আমার মনে পড়ছে না, সুজেতে নেই, এটা লিখলাম স্মৃতি থেকে, একটু দেখে নেবেন ম্যানপেজ পড়ে। স্ল্যাকে তো ওসব সোশাল ওয়ার্কের কোনো বালাই-ই নেই। আমার সুজে সিস্টেমের '/etc/mtab' আর '/proc/mounts' ফাইলদুটো তুলে দিই। পাশাপাশি। প্রথম যখন বুট হল, '/' আর '/boot' ছাড়া কেউ মাউন্ট নেই। তারপর, যখন অন্য চারটে পার্টিশনও মাউন্ট হয়েছে। যখন শুধু '/' আর '/boot' ডিরেক্টরির পার্টিশন দুটো, মানে '/dev/hdb3' আর '/dev/hdb1' মাউন্ট করা আছে, আর আছে সোয়াপ পার্টিশন —

ফাইল '/etc/mtab'	ফাইল '/proc/mounts'
/dev/hdb3 / xfs rw 0 0 proc /proc proc rw 0 0 devpts /dev/pts devpts rw,mode=0620,gid=5 0 0 /dev/hdb1 /boot xfs rw 0 0 shmfs /dev/shm shm rw 0 0 usbdevfs /proc/bus/usb usbdevfs rw 0 0	rootfs / rootfs rw 0 0 /dev/root / xfs rw 0 0 proc /proc proc rw 0 0 devpts /dev/pts devpts rw 0 0 /dev/hdb1 /boot xfs rw 0 0 shmfs /dev/shm shm rw 0 0 usbdevfs /proc/bus/usb usbdevfs rw 0 0

পরবর্তী চারটে ভৌত পার্টিশন '/dev/hda1', '/dev/hda5', '/dev/hdb5', '/dev/hda6' মাউন্ট হওয়ার পর দুটো ফাইলেই এল চারটে করে নতুন লাইন —

/dev/hda1 /mnt/windows/c vfat rw,nosuid,nodev 0 0
/dev/hda5 /mnt/windows/d vfat rw,nosuid,nodev 0 0
/dev/hdb5 /mnt/arkive reiserfs rw,nosuid,nodev 0 0
/dev/hda6 /mnt/slackware reiserfs rw,nosuid,nodev 0 0

এবং এই দুটো অবস্থায় '/proc/filesystems' ফাইলে কী বদল ঘটল দেখুন, পরের টেবিলে, যেখানে আমরা শুধু '/' আর '/boot' ডিরেক্টরির পার্টিশন দুটো মাউন্ট করা আছে এই অবস্থায় এবং তার পরে যখন সাতটা ভৌত পার্টিশনই মাউন্ট হয়ে আছে এই অবস্থায় ওই একই ফাইলের আকার দেখিয়েছি। যখন শুধু '/dev/hdb3' আর '/dev/hdb1' ডিরেক্টরির পার্টিশন মাউন্ট করা আছে তখন সিস্টেমে ব্যবহার হচ্ছে সতেরোটা ফাইলব্যবস্থা — rootfs bdev proc sockfs futexfs tmpfs shm pipefs ext2 ramfs minix iso9660 nfs devpts xfs usbdevfs usbfs — '/proc /filesystem' ফাইলসিস্টেমে এই সতেরোটার নাম আছে। আর, অন্য চারটে পার্টিশন '/dev/hda1', '/dev/hda5', '/dev/hdb5', '/dev/hda6' মাউন্ট হওয়ার পরে ওই তালিকায় যোগ হল আরো দুটো ফাইলব্যবস্থা — 'vfat' আর অন্যটা 'reiserfs' — কেন সেটা দেখুন, ভিফ্যাট মানে তো জানেন, গ্নু-লিনাক্স যে ভাবে উইনডোজ পার্টিশনকে চেনে, আর '/mnt/arkive' ডিরেক্টরির পার্টিশন '/dev/hdb5' এবং '/mnt/slackware' ডিরেক্টরির পার্টিশন '/dev/hda6' — এই দুটোতেই আছে রাইজারএফএস ফাইলব্যবস্থা, '/etc/fstab' ফাইল থেকে মিলিয়ে নিন। তার মানে দেখুন, '/proc' ডিরেক্টরির দুটো ফাইল '/proc/mounts' এবং '/proc/filesystems', আর '/etc' ডিরেক্টরির ফাইল '/etc/mtab' — এই মোট তিনটে ফাইল বদলাতে থাকছে মোট কী কী পার্টিশন মাউন্ট হচ্ছে এবং সেখানে কী কী ফাইলব্যবস্থা আছে তার উপরে নির্ভর করে। বদলানোর অন্য উৎসও থাকে, ওই যে সিস্টেমের বানানো ভার্চুয়াল ফাইলব্যবস্থাগুলো খেয়াল করুন, ওগুলোরও নিজেদের ভূমিকা আছে। যাকগে ও জটিলতায় গিয়ে আর কাজ নেই। এই ফাইলসিস্টেম সংক্রান্ত আর একটা ফাইলও থাকতে পারে '/etc' ডিরেক্টরিতে। তার নাম '/etc/mtools.conf'। এমটুলস নামের একটা প্যাকেজের কনফিগারেশন ফাইল। `mtools` কমান্ডটা লাগে এমএসডস (MS-DOS) ফাইলব্যবস্থায় কাজ করতে। ম্যানুয়াল পাতায় দেখুন, খুঁটিনাটি পেয়ে যাবেন।

ঠিক ফাইলসিস্টেমের সঙ্গে যুক্ত নয়, কিন্তু ফাইলের সঙ্গে ভীষণভাবেই জড়িত, এরকম আর একটা ফাইলের কথা উল্লেখ করি এই '/etc' ডিরেক্টরি থেকে, তার নাম ম্যাজিক (/etc/magic)। ছয় নম্বর দিনের ২.৫ নম্বর সেকশনের একদম শেষে গিয়ে দেখুন, আমরা এই প্রসঙ্গটা এনেছিলাম, যে ফাইলনাম এক্সটেনশনটা তো গ্নু-লিনাক্স তথা যে কোনো ইউনিক্স সিস্টেমেই আসলে মানববোধ্য রকমে ব্যবহার হয়, আপনি আমি আমরা বিভিন্ন জাতের এক্সটেনশন ব্যবহার করি, ঠিক পদবী যেমন, এক বর্গের ফাইলকে অন্য এক বর্গ থেকে আলাদা করতে। সিস্টেমের তাতে কিছু এসে যায়না। কিন্তু সিস্টেম তাহলে এক জাতের ফাইলকে আর এক জাতের ফাইল থেকে আলাদা করে কী করে? কোনোটা এক্সিকিউটেবল ফাইল কোনোটা ছবি কোনোটা টেক্সট কোনোটা ডেটাবেস ইত্যাদি। তার জন্যে একটা কমান্ডের কথা বলেছিলাম আমরা, যার নাম 'file'। '/etc/magic' নামের ফাইলের গোড়াতেই দেখুন লেখা আছে, পরপর দুই লাইনে, "# Magic data for file(1) command. # Format is described in magic(5)."। এর অর্থটাও আমরা জানি, আমাদের ব্রাউজনীয় ম্যানসমগ্রের এক নম্বর সেকশনে পাব 'file' কমান্ডটা যার ডেটা বা তথ্য হিশেবে ব্যবহৃত হয় এই '/etc/magic' ফাইলটা, এর বৃত্তান্ত আবার পাওয়া যাবে ওই সমগ্রেরই পাঁচ নম্বর সেকশনে। এক নম্বর সেকশনে, 'file.1.html'-এর গোড়াতেই দেখুন, বলা আছে, "File tests each argument in an attempt to classify it. There are three sets of tests, performed in this order: filesystem tests, magic number tests, and language tests. The *first* test that succeeds causes the file type to be printed."। তার মানে দেখুন, পাঁচ নম্বর সেকশনে লেখার সময়ে আমি ভুল জানতাম। লিখেছিলাম শুধু ম্যাজিক-সংখ্যার কথা, কিন্তু আসলে 'file' কমান্ডটা কোনো একটা ফাইলকে, যা তাকে কর্ম বা আর্গুমেন্ট হিশেবে দেওয়া হয়, তার চরিত্র বোঝার জন্যে পরখ করে দেখে তিনটে উপায়ে। কোনো একটায় উত্তর পেয়ে গেলে আর পরের টেস্টে যায়না। এই তিনটের প্রথমটা হল ফাইলসিস্টেম টেস্ট, দ্বিতীয়টা সেই ম্যাজিক নাম্বার টেস্ট, আর তৃতীয়টা হল ভাষা বা ল্যাংগুয়েজ টেস্ট। এর দু-নম্বরটাই আমি জানতাম এতদিন, অন্যদুটো এইমাত্র জানলাম, সাধে তখন ওইসব ভুলভাল লিখেছিলাম। তবে, একটা

জায়গায় মজা পাচ্ছি, এই ভুলটা কেউ ধরিয়ে দেয়নি, আমাদের এই বইটা বানিয়ে তোলার প্রক্রিয়ার মধ্যেই নিজে নিজে আবিষ্কৃত হয়ে গেছে। আমাদের এই পদ্ধতিটা কেমন কাজের দেখছেন — এটাই সেই জগতজুড়ে সুপ্রসিদ্ধ আরটিএফএম বা রিড-দি-ফাকিং-ম্যানুয়াল পদ্ধতি। এবার ম্যানসমগ্রের পাঁচ নম্বর সেকশন থেকে 'man.5.html'-এর সঙ্গে মিলিয়ে মিলিয়ে পড়ে গোটাটা শিখে ফেলুন। আপনি আর ম্যানসমগ্র নিজেদের মধ্যে বুঝে নিন, আমি আর নেই। গোবর্ধন আর হর্ষবর্ধনের মধ্যে বারকয়েক ধার-বিনিময় করার পরে শিব্রাম যেমন বলে দিয়েছিল, আপনারা নিজেরাই বুঝে নিন — আমি আর নেই। দেখছেন তো, কী সব ভুলভাল লিখছি, লোকে দেখলে কী বলবে?

২.৩.৩।। সিস্টেম-অ্যাডমিনস্ট্রেশন বা নিয়ন্ত্রণের কনফিগারেশন ফাইল

এই এলাকাটাও আমাদের অচেনা নয়, এর আগেও ফাইলগুলো নিয়ে বারংবার আলোচনা এসেছে। '/etc/group', '/etc/login.defs', '/etc/passwd', '/etc/securetty', '/etc/shadow', '/etc/shells', '/etc/motd' ইত্যাদি। এর মধ্যে কিছু কিছু ফাইল আমরা ইতিমধ্যেই চিনি, একদম ভেঙে ভেঙে আলোচনা করেছি, পাঁচ ছয় সাত আট নম্বর দিন জুড়ে। একটা কথা বলি, সব ডিস্ট্রোতেই আপনি যে হুবহু একই নামে একই চেহারায় একই কনফিগারেশন ফাইল পাবেন তা নয়, ডিস্ট্রো থেকে ডিস্ট্রোয় কিছুটা তফাত হয়ই। তবে সেসব শালগ্রামের ওঠাবসার মত, একটু এপাশ আর ওপাশ, একটু নাড়াচাড়া করতে থাকলেই দেখবেন গোটাটা আপনার কাছে স্পষ্ট হয়ে উঠছে।

'/etc/group' ফাইলকে আমরা চিনি। তিনি লিখে রাখেন সিস্টেমে উপস্থিত গ্রুপগুলোর নাম। আর কোন বা কোন কোন কোন ইউজার কোন বা কোন কোন গ্রুপে আছে সেই তথ্য। একজন ইউজার একাধিক রকমের কাজের সূত্রে একাধিক রকমের অনুমতির সূত্রে একাধিক গ্রুপে থাকতেই পারে। তার গোটাটাই লেখা থাকবে এই ফাইলে। নিজে পড়ে দেখুন ফাইলটা। এবং এর ম্যানপেজও।

এরকম আর একটা হল '/etc/login.defs' ফাইল। এর মধ্যে থাকে, এর মধ্যে থাকে লগিন (login) প্রোগ্রামের কনফিগারেশনে ব্যবহৃত বিভিন্ন সংজ্ঞা, এর নিজের ভাষায়, "Configuration control definitions for the login package", সহজ কথায় লগিন প্রোগ্রামের কনফিগারেশন ফাইল, এই প্রোগ্রামটা নিয়ে আমরা অনেক আলোচনা করেছি পাঁচ নম্বর দিনে, মনে করুন। ভুলে গিয়ে থাকলে '`man 1 login`' করে পড়ে নিন, বা আমাদের বানানো ওই ম্যানসমগ্রের এক নম্বর সেকশনে পাবেন, 'man.1.html'। আমরা '`set`' কমান্ড দিয়ে যে পাতার পর পাতা জুড়ে বিভিন্ন সিস্টেম ভ্যারিয়েবলের যে তালিকা পেয়েছিলাম, তার কয়েকটা দেখুন এই ফাইলেই দেওয়া। যেমন একটা, এখানেই দেওয়া, '`ENV_PATH /usr/local/bin:/usr/bin:/bin`'। এটা হল ডিফল্ট পথনির্দেশ বা পাথ-সেটিং। পাসওয়ার্ড সংক্রান্ত অনেক নিয়মকানুন, সিস্টেম যা মেনে চলে, দেওয়া থাকে এখানেই। সিস্টেমে ইউজারদের পরিচিতি (uid) সংখ্যা কত থেকে কতর মধ্যে থাকবে সেটাও দেওয়া থাকে এখানেই। ইত্যাদি।

'/etc/passwd' এবং '/etc/shadow' আমাদের বহুবার আলোচনা করা ফাইল। খুব দীর্ঘ আলোচনা আছে পাঁচ এবং সাত নম্বর দিনে, যদি ভুলে গিয়ে থাকেন, যা খুবই অস্বাভাবিক, একবার ফের দেখে আসুন। '/etc/passwd' ফাইলে থাকে ব্যবহারকারীদের বিষয়ে তথ্য, তাদের পাসওয়ার্ড বা প্রবেশসংকেত সহ, যদি-না তারা ছায়াবৃত থাকে, মানে যদি-না তাদের শ্যাডো পদ্ধতিতে রাখা হয় সিস্টেমে, আর সেই অবস্থায় সেই পাসওয়ার্ডগুলো থাকে '/etc/shadow' ফাইলে।

'/etc/securetty' ফাইলে থাকে, তার নিজেরই ভাষায়, "`This file contains the device names of tty lines (one per line, without leading /dev/) on which root is allowed to login.`"। তারপরেই 'tty1' থেকে 'tty6' পর্যন্ত ছটা নাম দেওয়া, এবং সঙ্গে সঙ্গে 'vc1' থেকে 'vc6'। মনে করতে পারছেন তো, 'tty' মানে টেলিটার্মিনাল, আর 'vc' মানে ভারচুয়াল কনসোল, মানে ভৌতিক পট, 'Ctrl-Alt-F1' থেকে 'Ctrl-Alt-F6' করে আমরা যাদের মধ্যে নড়ে বেড়ানোর কথা বলেছিলাম। একবার পাঁচ নম্বর সেকশন থেকে 'securetty' ফাইলের ম্যানপেজ পড়ে নিন।

'/etc/shells' ফাইলে থাকে সিস্টেমে প্রাপ্তব্য শেলের তালিকা তাদের পথনির্দেশ সহ। কিন্তু আমার মেশিনের সুজে সিস্টেমের এই তালিকায় দেওয়া নাম গুলো তুলে দি, একটা মজা আছে এখানে। '/bin/ash /bin/bash /bin/bash1 /bin/csh /bin/false /bin/ksh /bin/sh /bin/tcsh /bin/true /bin/zsh /usr/bin/csh /usr/bin/ksh /usr/bin/passwd /usr/bin/bash /usr/bin/rbash /usr/bin/tcsh /usr/bin/zsh'। এদের মধ্যে কতকগুলো নাম তো ইতিমধ্যেই আমরা

শেল হিশেবে জেনে গেছি। কিন্তু এর মধ্যে ‘passwd’ বা ‘true’ বা ‘false’ — এর মানে কী? দেখুন, আমার আর এনার্জি নেই, এই নয় নম্বর দিনটাই শেষ হচ্ছেনা, এর পরে তো আবার একটা দশ নম্বর দিন আছে, শেল স্ক্রিপ্ট নিয়ে, অত জ্ঞানপিপাসা হলে নিজে খুটে খান।

‘/etc/motd’ ফাইলে থাকে আপনার বাণী, যদি আপনি আপনার সিস্টেমে সাধারণ ব্যবহারকারীদের, নিজেকে ধরে, লগ-ইন করা মাত্র কোনো বাণী দিতে চান, তাহলে সেই বাণীর পাদপীঠ হল এই ফাইল। সিস্টেম অ্যাডমিনস্ট্রেটরের ‘message-of-the-day’ বা ‘দিবস-পিছু-বাণী’ ধরা থাকে এই ‘/etc/motd’ ফাইলে। বাণীদানের অত শখ যদি থাকে নিজে খুঁজে বার করে পড়ে নিন, কোন সেকশনে পাবেন এর ম্যানুয়াল সেটা মনে না-করতে পারলে বুঝতে হবে আপনার এখনো বাণীবন্দনার দিন চলছে, বাণীপ্রদানের বয়েস হয়নি।

২.৩.৪।। সিস্টেম কমান্ডের কনফিগারেশন

সিস্টেম কমান্ডদের দিয়ে সিস্টেমকে নিয়ন্ত্রণ করা হয়, সব কিছুকে ঠিকঠাক রাখা হয় তার নিজের নিজের জায়গায়, তরকারির বাটি ড্রেসিংটেবিলে আর ডিপফ্রিজে সাবান এই অবস্থাটা যাতে না হয়। যেমন, পাঁচ নম্বর দিনে আমাদের লগ-ইন নিয়ে আলোচনাটা মনে করুন, তার কাজের একদম শেষে এসে সে তার হাতের ব্যাটন দিয়ে গেল ব্যাশকে, এবং ব্যবহারকারীকে সংস্থিত করে দিল তার নিজের ঘরে, এবার এখানে বসে তুই যা করবি কর, নিজের পাঁঠা তুই লেজে কাটবি না শিঙে সেটা তোর নিজের অভিলাষ। আমি যেমন আমার হোম-এ বসে একবার রেকার্সিভলি মানে যতগভীর অব্দি ডিরেক্টরির ছানার ছানা থাকুক না কেন সেই সমস্ত ফাইল সবার জন্যে পাঠযোগ্য লিখনযোগ্য এবং চালনীয় করে দিয়েছিলাম, কমান্ডটা দিয়েছিলাম ‘`chmod -R 777 *`’। এতে কিন্তু ডটানন ফাইলদের ধরা যায়না, ব্যাশ ‘`*`’ চিহ্নে তাদের ধরেনা, সেইজন্যে তাদেরও আলাদা করে, ‘`chmod -R 777 .*`’। এতে আর কিছুই হয়নি, শুধু আমায় নতুন করে একটু ইনস্টল করতে হয়েছিল, তখন তো এত লাটকে লাট ব্যাকআপও থাকত না। ও, এটা কিন্তু একটা সিস্টেম ঘেঁটে দেওয়ার একটা উৎকৃষ্ট উপায়, ‘`chmod -R -x /`’। এতে সব ফাইলই থাকবে, সবই থাকবে, শুধু কোনোকিছুই আর করতে পারবে না। উইনডোজে যেমন ভারি কাব্যিক রকমের খার মেটানোর একটা কায়দা হল কারোর মেশিনে বসে তার রেজিস্ট্রিটাকে অন্য কোনো নামে রপ্তানি করে দেওয়া। তখন কোনো হেলদোলই হবেনা, বোঝা যাবে পরের বার রিবুট করার সময়, মানে, রিবুটনা-করার সময়।

লগ-ইন তার ব্যাটন যার হাতে দিয়ে গেল, সিস্টেম আর ব্যবহারকারীর মধ্যে সেই নিরলস দোভাষী মানে ব্যাশ, তিনিও একটি সিস্টেম কমান্ড। ব্যাশের কনফিগারেশন তথা ব্যাশভূষা নিয়ে আমরা কথা বলব পরের দিন, হায়, আবার একটা দিন, এখন অন্য সিস্টেম কমান্ডদের কনফিগারেশনের আলোচনায় আসা যাক। এই ফাইলগুলোকে একটু বুঝে রাখা বেশ জরুরি, শুধু সিস্টেমকর্তা বা রুট না, এমনি ব্যবহারকারীরও। আর আমাদের অনেকেরই একা-একলা একক-ব্যবহারকারীর পিসির বেলায় তো সেই সোহম মন্ত্র — সেই রুটই আমি, আমিই পরম ব্রহ্ম। এই ফাইলগুলোর মধ্যে বিশেষ কয়েকটা হল ‘/etc/lilo.conf’, ‘/etc/logrotate.conf’, ‘ld.so.conf’, ‘/etc/inittab’, ‘/etc/termcap’ ইত্যাদি। এর মধ্যে কিন্তু ডিস্ট্রো থেকে ডিস্ট্রো থেকে কিছু তফাত থাকে। যেমন ‘identd.conf’ স্ল্যাকওয়ারে আছে কিন্তু সুজেতে নেই। আবার ‘termcap’ সুজেতে আছে, তবে অন্য জায়গায়, ‘/usr/share/misc’ ডিরেক্টরিতে। এইগুলো নিজেই দেখে অভ্যস্ত হয়ে নিতে হবে। ‘/etc’ ডিরেক্টরির সব ফাইলগুলো একটা ফাইলে লিখে নিয়েছিলাম, ‘writefiles’ নামে একটা স্ক্রিপ্ট দিয়ে, সেরকম অনেকগুলো ডিরেক্টরিতেই করে নিন, তারপর তাদের পেজ ডাউন করে যান, একধরনের একটা চেনাজানা তৈরি হয়ে যাবে, যেটা পরে কাজে লাগবে। আর আমার মত যদি সমাজসচেতন ডটম্যাট্রিক্স প্রিন্টার হয়, প্রিন্ট আউটও বার করে নিতে পারেন, অন্তত ‘/etc’ ডিরেক্টরির টেক্সট ফাইলটার।

এর মধ্যে ‘/etc/lilo.conf’ আর ‘/etc/inittab’ আমরা চটকে রেখেছি আগেই, একাধিকবার করে, পুরো লাইন তুলে তুলে। কিন্তু এবার যদি একবার ওগুলো পড়েন, আর যাদের জন্যে এই বই লেখা আপনি যদি তাদের একজন হন, মানে অন্তত তথাগত সঙ্কর্ষণ অরিজিতের মত কোমর-বাগিয়ে-খুঁত-ধরতে-বসা মর্ষকামানন্দ না-হন, তাহলে নতুন করে পড়তে গিয়ে আপনি দেখবেন, অন্তত আগের বারের চেয়ে একটু বেশি বুঝতে পারছেন-ই। আমাদের সব চেনাগুলোই এইরকম, একই জায়গায় ফিরে ফিরে আসা বারবার, কিন্তু গতিপথটা হল অধিকেন্দ্রিক অধিত্যকা (কার্টসি সঙ্কর্ষণের ‘এপিসেন্ট্রিক র‍্যাভিনস’) যা ভাঙাচোরা পাথরে ভর্তি, তাই প্রতিবার একই হাঁটা আসলে নতুন নতুন হাঁটার জন্ম দিতে থাকে। শুরু করা যাক ‘/etc/logrotate.conf’ ফাইল দিয়ে। এর শুরুতেই নিদান দেওয়া, ঠিক যেকথা আমিও এতবার

বলে চলেছি, ম্যান পড়ার নেশা করো ম্যান, “`see "man logrotate" for details`”, তার পরের লাইনেই লেখা, “`rotate log files weekly`”। এই ‘/etc/logrotate.conf’ ফাইলটার কাজ কী সেটা আট নম্বর সেকশনে লগরোটেট-এর ম্যানুয়াল থেকেই পড়া যাক, “logrotate is designed to ease administration of systems that generate large numbers of log files. It allows automatic rotation, compression, removal, and mailing of log files. Each log file may be handled daily, weekly, monthly, or when it grows too large.”। গ্নু-লিনাক্স সিস্টেমের ‘/var/log’ ডিরেক্টরিতে প্রতিমুহূর্তে নানা লগ-ফাইল জমা হচ্ছে, একবার ঢুকে দেখুন। এই লগফাইলগুলো জমতে জমতে জগদ্দল যাতে না-হয়ে যায় তার জন্যে সিস্টেম নিজেই সরিয়ে ফেলার, ঘোরানোর, এবং কুঁকড়ে ফেলার কিছু কাজ করে চলে দিনপ্রতি বা হপ্তাপ্রতি। সেই কাজটার কনফিগারেশন লেখা থাকে এই ‘/etc/logrotate.conf’ ফাইলে।

‘/etc/ld.so.conf’ হল ডায়নামিক লিংকারের বা গতিশীল সংযোগকারীর কনফিগারেশন ফাইল। কিন্তু, তার মানে কী? আমরা লাইব্রেরি ফাইলের কথা আলোচনা করেছি মনে আছে? একটা প্রোগ্রামিং ভাষায় লেখা মানববোধ্য প্রোগ্রাম, যা কম্পাইল করে একটা চালনীয় বা এক্সিকিউটেবল ফাইল মানে প্রোগ্রাম তৈরি হয়, তার মধ্যে কিছু ফাংশন থাকে সার্বজনীন, যা যে কোনো প্রোগ্রাম ব্যবহার করে, যাকে আমরা ডেকেছিলাম লাইব্রেরি ফাংশন বলে। এবার, কোনো একটা বিশেষ লাইব্রেরি ফাংশন একাধিক প্রোগ্রামের জন্যেই ব্যবহার হতে পারে। তখন সেই লাইব্রেরিটাকে শেয়ার করাতে হয়, মানে একাধিক প্রোগ্রামের মধ্যেই ব্যবহার করাতে হয় এই গতিশীল সংযোগ বা ডায়নামিক লিংক দিয়ে। নয়তো স্ট্যাটিক বা স্থির রকমেও বাইনারি তৈরি করে নেওয়া যায়, যা প্রতিটা বাইনারির অভ্যন্তরে ব্যবহৃত প্রতিটি লাইব্রেরিকে রেখে দেয়। এতে বাইনারির সাইজ খুব বেড়ে যায়। এবার ‘ld.so’ বা ‘ld-linux.so’ হল সেই ডায়নামিক লিংকার, এবং এরই কনফিগারেশন ফাইল হল ‘/etc/ld.so.conf’। ভালো করে বুঝতে হলে একটু প্রোগ্রামিং এবং কম্পাইলেশনের ধারণা লাগবে, আর ম্যানপেজের আট নম্বর সেকশন থেকে ‘ld.so’, ‘ldd’, ‘ldconfig’ — এগুলোর ম্যানপেজ পড়ে ফেলুন। ফাইলটাতে দেখুন সমস্ত বারোয়ারি বা শেয়ারড লাইব্রেরিগুলোর ঠিকানা দেওয়া আছে।

‘/etc/termcap’ ফাইলে দেওয়া থাকে সমস্ত টার্মিনালের তথা ক্যারেকটার ডিভাইসের হালহকিকত। এটা পুরোনো প্রথা, এখনো সিস্টেমে দেওয়া আগেকার প্রোগ্রামগুলোর সঙ্গে সাযুয্য বা কম্পেটিবিলিটি রাখার জন্যে। এখন মূলত ‘termcap’-এর জায়গায় ‘terminfo’ ব্যবহার হয়। এই দুটোর সঙ্গে ‘term’-এরও ম্যানুয়াল পেজ পড়ে নিন। তিনটেই পাবেন পাঁচ নম্বর সেকশনে। আচ্ছা এর আগেই পরপর বেশ কয়েকটা ম্যানপেজ এসেছে সেকশন আট থেকে। আট নম্বর সেকশনে কী ধরনের ম্যানুয়াল থাকে বলুন তো?

অশেষের অলংকার দিয়েই শুরু করেছিলাম আজকের আলোচনা, সেই বাঁশ দিয়েই শেষ করা যাক, এর পরের মানে শেষের দিন শুরু করব ব্যাশ দিয়ে, তারপরেই ব্যাস। এই গোটা আলোচনার সিরিজে, জিএলটি ইশকুল পাঠমালায়, অশেষের অলংকারে, এটাই দীর্ঘতম বাঁশ, এই কথাশেষটুকু বাদ দিয়েই ১৭২৪৮ শব্দ, এর আগে সবচেয়ে দীর্ঘ ছিল সাত নম্বর দিন, ১৬১৭৪ শব্দ। আমি নিজে আর পারছিলাম না, জানিনা আপনাদের কী অবস্থা, এরপরে বাঁশগাছে উঠে বসবাস করতে না-হয়। তাও তো যখ মানে ডিমনগুলোর আলোচনা ছেড়ে দিলাম, তার ব্যাশ তথা অন্যান্য ইউজার প্রোগ্রাম কনফিগারেশন রেখে দিলাম পরের মানে শেষের দিনের জন্যে।

glt-mad@ilug-cal.org

MADHYAMGRAM

সংকলন ও রচনা : মধ্যমগ্রাম জিএলটি-র (glt-mad@ilug-cal.org) তরফে : ত্রিদিব সেনগুপ্ত

গ্নু-লিনাক্স ইশকুল

জিএলটি মধ্যমগ্রাম
glt-mad@ilug-cal.org

আজকের আলোচনা ব্যাশ শেল নিয়ে। প্রথমে তার কনফিগারেশন, সেখান থেকে ব্যাশের কাজ করার রকম নিয়ে কিছু কথা। কিছুটা আলোচনা, একটা ছবি সহ, শেল কী ভাবে আপনার আদেশ পালন করে, আপনার এবং কারনেলের ভিতর সংযোগ রাখে, আমরা করেছিলাম পাঁচ নম্বর দিনে। তখন সিস্টেমের বহু খুঁটিনাটি আমরা জানতাম না, সেই আলোচনাটাকেই এবার আমরা বাড়িয়ে নিয়ে যাব। কিছু কিছু শেল স্ক্রিপ্ট এবং শেলকে দিয়ে কী করে অনেকটা বা অনেক রকম কাজ একই সঙ্গে করা যায় তাই নিয়ে কিছু প্রসঙ্গ আমরা সাত এবং নয় নম্বর দিনেও এনেছি, সেই আলোচনাগুলোকে এবার আমরা আরো অনেকটা অব্দি নিয়ে যাব। আপনার সিস্টেমের ভিতরেই প্রচুর ডকুমেন্টেশন আছে ব্যাশ নিয়ে, তারপরেও আছে লিনাক্স ডকুমেন্টেশন প্রোজেক্ট মানে 'www.tldp.org' বা আছে ফ্যাক অর্গানাইজেশন মানে 'www.faq.org'। এবং এছাড়াও আরো প্রচুর প্রচুর ডকুমেন্টেশন ছড়িয়ে আছে নেটে। ব্যাশ শেল বা ব্যাশ স্ক্রিপ্ট সার্চে দিয়ে যদি একবারো গুগলি মেরে, 'www.google.com', শখানেকের কম লিংক পান, ফ্রিতে আমায় কেলিয়ে যেতে পারেন। এরকম প্রচুর মালপত্তর আমার কাছে নামানো আছে, আর আমি মানুষ হিশেবে অত্যন্ত উদার হওয়ায় আপনি যদি আমার সঙ্গে বসে সেসব পড়তে চান, আমার কোনো আপত্তি নেই। তবে মনে রাখবেন, নিখিল-ভারত-আইনপন্থী-হায়ার-কমিটির চেয়ারম্যান পদ পাওয়ার কথা চলছে আমার, তাই কোনো কপিরাইটরিক্ত জিনিষপত্তর আমার কাছ থেকে সিডিতে লিখে নেওয়ার আশা করবেন না। এমনকি আমার বার্নার ড্রাইভটা অব্দি কোনো বেআইনি জিনিষ কপি করতে দিলে কাজ থামিয়ে জনগনমন গাইতে শুরু করে। সোজা হয়ে দাঁড়িয়েও নেয়, রবীন্দ্রপ্রণামের মূল উদ্দিষ্ট ব্রিটিশ সম্রাট পঞ্চম জর্জ থেকে আজকের ভারতীয় সংবিধান অব্দি সকলের প্রতি বিনম্র শ্রদ্ধায় — মানে, শ্রদ্ধয়া দেয়ম, শ্রাদ্ধ পালন করে।

।।দিন দশ।।

১।। ব্যবহারকারীর প্রোগ্রামের কনফিগারেশন ফাইল

সাত নম্বর দিনের ২.৩ সেকশনে ব্যাশের কাছে তুলে রাখা আপনার আগে-ব্যবহৃত আদেশের ইতিহাসের প্রসঙ্গে আমরা এই আলোচনাটাকে একটু এনেছিলাম। আমরা একটা হোম ডিরেক্টরির মোট ডটানন ফাইল এবং আরসি (rc) ফাইলের একটা অসম্পূর্ণ তালিকা দিয়েছিলাম। এবার সেই ফাইলগুলোকে একটু বোঝার চেষ্টা করা যাক। তার আগে জানা দরকার, ব্যবহারকারীর প্রোগ্রাম বলতে আমরা কী বুঝছি। বাইনারি এবং সিস্টেম বাইনারির আলোচনায় আমরা রুটের সঙ্গে অন্য ব্যবহারকারীর অধিকারের তফাতের কথা বলেছিলাম। এছাড়া গ্লোবাল এবং লোকাল সেটিং-এর প্রসঙ্গটাও এখন আমাদের অপরিচিত নয়। খুব ছোট অর্থে আমরা লোকাল বলতে ধরব একজন নির্দিষ্ট ব্যবহারকারীর বেলায় যা সত্যি, কিন্তু সিস্টেমের যে কোনো ব্যবহারকারীর বেলায় নয়। ধরুন আমি একজন সাধারণ ব্যবহারকারী হিশেবে আমি এমন ব্যবস্থা করতেই পারি, সত্যিই এই রকম ব্যবস্থা করা আছে যে, আমি লগ-ইন করা মাত্র আমার শেল আমায় একটা ছোট্ট দেখে আপত্তিকর মানে অফেনসিভ বাণী শোনায় ফরচুনের। অফেনসিভ ফরচুন মানে কতটা অফেনসিভ সেটা সুরুচি এবং কালচারের সমস্ত পিছুটান থেকে সম্পূর্ণ মুক্ত না-হলে সত্যিই দেখতে যাওয়ার রিস্ক নেওয়া উচিত নয়, আমার বন্ধুরা জানেন যে গর্ব আমি অনায়াসে করতে পারি। বেশিরভাব ডিস্ট্রোই অফেনসিভ ফরচুন নিজের মধ্যে দেয়না, আলাদা করে ডাউনলোড করে ইনস্টল করে নিতে হয়। এবার দেখুন, এই '`fortune -os`' ব্যবস্থাটা, মানে সেই সব ফরচুন বাণী যারা শর্ট এবং অফেনসিভ, ছোট্ট এবং আপত্তিকর, শোনানোর ব্যবস্থাটা আমার মানে ইউজার 'dd'-র নিজের হোম ডিরেক্টরিতে করা আছে আমার ব্যাশের কনফিগারেশনে। কিন্তু এটা ইউজার 'manu' বা 'piu' বা 'atithi'-র কনফিগারেশনে করা নেই। তাই এটা সেই অর্থে লোকাল সেটিং। এরকম ইউজার প্রোগ্রামেও করা আছে। আমি ইম্যাক্স খুললেই সেটা আমার নিজের কিছু পছন্দ মোতাবেক চলে, সেটাও লোকাল সেটিং। এরকম অনেক প্রোগ্রামেই আছে। আবার ধরুন, আমি সেদিন ভিম (vim) সফটওয়ারটার একটা নতুন ভার্শন, ভার্শন ৬.২, যেটা সুজে ৮.২-এ দেওয়া ছিলনা, নামিয়েছিলাম নেট থেকে, 'vim-6.2.tar.bz2' ফাইলে এবং সেই কোঁকড়ানো সিন্দুকটাকে '/opt' ডিরেক্টরিতে রেখে '`tar xvjf vim-6.2.tar.bz2`' কমান্ড দিয়ে ভেঙে, নতুন গজানো '/opt/vim-6.2' ডিরেক্টরিতে গিয়ে '`./configure && make && make install`' করে

ইনস্টল করেছিলাম। (এই যে, ইনস্টল করার কায়দাটা অকারণেই একবার বলে নিলাম, এটা কিন্তু আপনাকে মনে পড়ানোর জন্যে যে গোটাটা আপনার মনে পড়ছে কিনা, কোথায় এই আলোচনাটা এসেছিল, মনে করুন তো।) এই গোটাটাই করেছিলাম রুট হয়ে, কারণ এটা ছিল আমার মেশিনের সুজে ৮.২ সিস্টেমে গ্লোবাল ইনস্টল। এই নতুন ভিমটা এবার আমার সিস্টেমের প্রত্যেক ব্যবহারকারীই পাবে, সেই অর্থে এটা আমার সিস্টেমের গ্লোবাল সেটিং। কিন্তু এই 'লোকাল-গ্লোবাল' বাগধারাটা আদতে কোথা থেকে এসেছিল সেটাও মাথায় রাখবেন, একটা নেটওয়ার্কিত সিস্টেমে একটা মেশিন হল লোকাল, গোটা নেটওয়ার্কটা গ্লোবাল। একজন সাধারণ ব্যবহারকারী শুধু লোকাল ইউজার প্রোগ্রামগুলোই চালাতে পারেন। লোকাল বা গ্লোবাল কোনোরকম সিস্টেম প্রোগ্রামই তার অধিকারের মধ্যে পড়েনা, এমনকি গ্লোবাল ইউজার প্রোগ্রামও না, সেগুলোর জন্যেও গোটা সিস্টেমব্যাপী কিছু বদল এবং কনফিগারেশনের দরকার পড়ে তা তার অধিকারের মধ্যে নেই। তার অধিকার তার হোম ডিরেক্টরিটুকু। এমনকি সেখানেও রুট যে কোনো বদল যে কোনো সময়েই ঘটাতে পারে।

ইউজার বা সিস্টেম দুরকম প্রোগ্রামই তাদের চালানো মাত্র নিজেদের কনফিগারেশন ফাইল একবার পড়ে নেয়। কোনো কোনো সিস্টেম প্রোগ্রাম চালু হয় একদম সিস্টেম বুট করার সময়েই, তাই তাদের কনফিগারেশন ফাইলও পড়া হয় সেই সময়েই। '/etc' ডিরেক্টরির সেইসব কনফিগারেশন ফাইল নিয়ে আমরা আলোচনা করেছি। ইউজার প্রোগ্রামগুলোরও একটা করে কনফিগারেশন ফাইল রাখা থাকে বা থাকতে পারে '/etc' ডিরেক্টরিতেই। প্রোগ্রামটা চালু হওয়ার সময় সেই ফাইল পড়ে নেবে। সেই ফাইল থেকে প্রোগ্রামটা পড়ে নেবে ডিফল্ট কনফিগারেশন। তারপরে, ব্যবহারকারী যদি নিজের কোনো বদল আনতে চায় সেটা ঘটাবে তার নিজের হোম ডিরেক্টরিতে রাখা বাড়তি একটা ব্যক্তিগত কনফিগারেশন ফাইলে। এই ফাইল মোতাবেক বদলটা দেখা দেবে শুধু ওই প্রোগ্রামের ওই ব্যবহারকারীর ব্যক্তিগত বা লোকাল ব্যবহারেই। প্রোগ্রামটার গ্লোবাল ব্যবহার নিয়ন্ত্রিত হবে '/etc' ডিরেক্টরির ওই গ্লোবাল কনফিগারেশন দিয়ে। আর যদি প্রোগ্রামটা গ্লোবালি ইনস্টল না-করে, শুধু একক ব্যবহারকারীর জন্যে লোকালি, তার হোম ডিরেক্টরিতেই ইনস্টল করা হয়ে থাকে তাহলে তো চুকেই গেল। কোনো কোনো প্রোগ্রামে তা হয় বৈকি। ফন্টের বেলাতেও হয়।

ধরুন ইম্যাক্স। এর কনফিগারেশনের জন্যে গ্লোবালি দেওয়া আছে '/etc/skel/.gnu-emacs' বা '/etc/skel/.emacs'। দেখুন, দুটোই ডটানন ফাইল, শুরুতে একটা করে বিন্দু বা ডট বা '.' আছে। আর ব্যক্তিগত ব্যবহারকারী 'dd'-র হোম ডিরেক্টরিতে বা '/home/dd'-তে আছে '.gnu-emacs' বা '.emacs'। যা মূলত '/etc/skel' ডিরেক্টরির ওই গ্লোবাল কনফিগারেশন ফাইলদুটোরই কপি। এবার পিউ ইম্যাক্সে সি-প্রোগ্রাম লেখা বা কম্পাইল করার জন্যে কিছু সুবিধেজনক বদল এনেছিল, যা লিখে নিয়েছিল ওর হোম ডিরেক্টরির '/home/piu/.emacs' ফাইলে। সেটা আমার হোম ডিরেক্টরির '/home/dd/.emacs' ফাইলে না-থাকায় সেই ব্যাপারগুলো আমি ইম্যাক্স ব্যবহার করার সময় ঘটেনা। এবার ভাবুন ইম্যাক্স তাহলে কী করছে — যেই কেউ `emacs` মন্তর পড়ে ইম্যাক্স প্রোগ্রামটাকে জাগিয়ে তুলছে, ইম্যাক্স পড়ছে '~' ডিরেক্টরি থেকে ব্যক্তিগত কনফিগারেশন মানে '~/.emacs' ফাইল। এই '~' একটা চিহ্ন যা দিয়ে ব্যাশ শেল কোনো ব্যবহারকারীর হোম ডিরেক্টরিকে চিনে নেয়। তার মানে প্রোগ্রামটা যখন আমি মানে ইউজার 'dd' চালাচ্ছে তখন '~' হল '/home/dd'। আর যখন পিউ চালাচ্ছে, তখন '~' হল '/home/piu'। আর যদি কোনো ব্যক্তিগতভাবে বদলানো '~/.emacs' ফাইল না পায় তাহলে পড়ছে '/etc/skel' ডিরেক্টরি থেকে গ্লোবাল কনফিগারেশন। একটু বাদেই এইসব '~' জাতীয় খ্যাঁচগুলো ভালো করে বুঝতে হবে আমাদের।

আর একটা ধরুন আমার প্রিয় প্রোগ্রাম 'mplayer'। যাদুপর্বতের ওই আনুভূমিক আর উল্লম্বের রসিকতাটাকে একটু বদলে নিয়ে বলাই যায়, যখনই আমি অক্ষর দেখছি-না, মানে লিখছি বা পড়ছি না, তখনই আমি সিনেমা দেখছি। আর আমাদের তথাগত তো সোনার ছেলে, কী যে সব ভালো ভালো ফিল্ম আজকাল ওর কল্যাণে দেখতে পারছি আমি, সে আর বলার নয়। এবার এই 'mplayer' কাজ করে মূলত গ্লোবাল কনফিগারেশন ফাইল '/etc/mplayer.conf' দিয়ে, আর কোনো বাড়তি পছন্দ আপনি রেখে দিতে পারেন '~/.mplayer' ডিরেক্টরির '~/.mplayer/config' ফাইলে। এই '~/.mplayer/config' ফাইলটা আসলে '/etc/mplayer.conf' ফাইলেরই একটা কপি করে দিতে পারেন আপনি, তার পরে তাতে প্রয়োজনীয় পছন্দমত বদলগুলো ঘটালেন। দুটো ফাইলই যদি থাকে সিস্টেমে, এবং দুটোর সবকিছু না মেলে, তখন 'mplayer' মানবে '~/.mplayer/config' ফাইলে দেওয়া নিদানটাই। আর, শুধু এই দুটো জায়গাই না,

আর একটা তৃতীয় জায়গাও আছে কনফিগারেশন ফাইল রাখার, যে তিনটে জায়গাকেই খোঁজে 'mplayer', পরপর। এই তৃতীয় জায়গাটা হল '/usr/local/etc/mplayer.conf'। গ্নু-লিনাক্স ফাইলসিস্টেম হায়েরার্কি নিয়ে আমাদের আলোচনা থেকে দেখুন তো আপনি এই তৃতীয় জায়গাটার চক্করটা বুঝতে পারছেন কিনা। '/etc/mplayer.conf' ফাইলে পরপর দেওয়া আছে বিভিন্ন ধরনের সেটিং, শুধু তার শেষটা অনবদ্য, সেটা তুলে দেওয়ার লোভটা সামলাতে পারছি না। এই অংশটায় বলা আছে, শুধু উপরে দেওয়া ওই তিনটে জায়গাই নয়, যে কোনো জায়গায় যে কোনো ডিরেক্টরিতে আপনি কনফিগারেশন ফাইলটা রাখতে পারেন যদি চান, শুধু তার পুরো পথটা দিয়ে দিতে হবে, আর তখন এই ডিফল্ট ফাইলটা উড়িয়ে দিতে হবে, নয়ত 'mplayer' এই ডিফল্ট ফাইলটাই পড়বে। বলে একটু মিচকি হেসেছেও, মানে মিচকি হাসির টেক্সট প্রতিরূপটা ':)'। আর এরপরেই হল সেই অনবদ্য শেষ লাইন, যেখানে সেই পথটা লিখে দেখিয়ে দেওয়া আছে, আরটিএফএম শব্দটা মনে আছে তো?

```
## You can also include other configfiles
## Specify full path!
## Delete this default :)
#include = /home/gabucino/.mplayer/i_did_not_RTFM_carefully_enough...
```

ব্যবহারকারী বা ইউজারের স্তরে যে কোনো প্রোগ্রামই মোটামুটি প্রথমে খোঁজে '/etc' ডিরেক্টরির কনফিগারেশন ফাইল, সেখান থেকে পড়ে নেয় ডিফল্ট কনফিগারেশন। তারপরে ইউজার সেই প্রোগ্রামের কোনো কাস্টমাইজেশন বা নিজের পছন্দ অনুযায়ী বদল ঘটিয়েছে কিনা সেটা দেখে নেয় কোনো ডটানন ফাইলে বা আরসি ফাইলে দেওয়া কনফিগারেশন থেকে। এই আরসি ফাইল নিয়ে একটা মজা ঘটল। আমার আরসি বলতে মনে হচ্ছিল রিসোর্সের সঙ্গে কোনো একটা সম্পর্ক আছে, কিন্তু ঠিক মনে করতে পারছিলাম না, কোলকাতা লাগের মেইলিং লিস্টে, মানে, 'ilug-cal@ilug-cal.org' ইমেল আইডিতে, একটা চিঠি দিলাম, কেউ জানে কিনা, আরসি (rc) ঠিক কিসের অ্যাক্রোনিম। কিন্তু মেলটা পাঠানোর পরপরই 'www.faq.org' সাইটে এরিক রেমন্ডস-এর জার্গন ফাইলে পেয়ে গেলাম সংজ্ঞাটা। আমরা চার নম্বর দিনের ৭ নম্বর সেকশনে ১৯৬২-তে এমআইটির সিটিএসএস বা কম্পেটিবল টাইম শেয়ারিং সিস্টেমের কথা বলেছিলাম মনে আছে? তাতে সিস্টেম চালু হওয়ার সময়ে ব্যবহৃত রানকম (runcom) ফাইল থেকে এসেছে নামটা। এবং মেলিং লিস্টেই তথাগত লিখল, ও-ও আমারই মত, মনে করত, রিসোর্স কনফিগারেশন গোছের কোনো কথার অ্যাক্রোনিম এটা। এবং দেখুন, দুজনেই একই কথা ভাবছি মানে, যতদূর সম্ভব কোথাও একটা পেয়েছি নিশ্চয়ই। তা যাই হোক, সিটিএসএস থেকে আসা যে এই 'rc' নামটা এখন লাগু হয়ে গেছে যে কোনো বুটকালীন চলা স্ক্রিপ্টের জন্যেই, একটা কমন-নাউন হিশেবে।

সাধারন ব্যবহারকারী বা ইউজারের স্তরে এই ডট ফাইল এবং আরসি ফাইলের মূল প্রয়োজনীয়তার জায়গাটা কিন্তু রুট এবং ইউজারের অধিকারের পার্থক্য। প্রথমত, ন নম্বর দিনে আপনারা দেখেছেন, '/etc' ডিরেক্টরির বিভিন্ন কনফিগারেশন ফাইলের সঙ্গে বদলে ফেলার মত সহজ সম্পর্কে আসাটা শুধু শ্রমসাধ্যই নয়, যথেষ্ট গা-ছমছমেও বটে। বদলাব যে, কিছু যদি ঘেঁটে যায়? আর একটা সিস্টেমে প্রতিটা জিনিষই তো অন্য জিনিষগুলোর সঙ্গে অতিনির্ণয়ের ওভারডিটারমিনেশনের সম্পর্কে আছে, প্রতিটি কিছুই অন্য প্রতিটি কিছুর দ্বারা নির্ণীত এবং নির্মিত। তার মানে একটা কোনো বেমতলব বদল কতদূর অব্দি কোনটাকে ঘেঁটে দিতে পারে সেটা সম্পর্কে একটা আন্দাজ তৈরি হওয়ারও দরকার পড়ে। এবার ধরুন কোনো একটা প্রোগ্রামের একটা বিশেষ ব্যবহার আপনার পছন্দ হচ্ছেনা, আপনি সেটা বদলাতে চান, এবং জানেন যে সেটা বদলানো যায়, তখন খুব সহজেই যেটা করতে পারেন তা হল নিজের হোম ডিরেক্টরির কনফিগারেশন ফাইলটা বদলে নেওয়া, আর সেটা অনাবশ্যক ভাবে ঘেঁটে গেলে, আবার মূল '/etc' ডিরেক্টরির কনফিগারেশন ফাইলটা বদলানো ফাইলের জায়গায় পুনঃস্থাপিত করে দেওয়া। আরো একটা সমস্যা থাকে বহু মেশিনের বহু ইউজারের সিস্টেমে, সেটা এই যে, '/etc' ডিরেক্টরির ভিতরে কোনো ফাইলে হাত দেওয়ার অধিকার তো থাকবে না আপনার। ওটা গ্লোবাল সিস্টেম, তাই রুটের এলাকা। একা একলা একটা পিসিতে অবশ্য এই ঝামেলাটা থাকেনা। এই সব কারণেই এই দুই সেট ফাইল, একটা গ্লোবাল কনফিগারেশন, '/etc' ডিরেক্টরিতে, অন্যটা লোকাল, ইউজারের হোমে, ডট ফাইল বা আরসি ফাইল। এই ফাইলগুলোর বিষয়ে সামগ্রিকভাবে আর কোনো আলোচনায় যাচ্ছিনা, কারণ কিছু ফাইলের কথা তো বললাম, এরকম নামের মিল থেকে বুঝে নিতে পারবেন, আর অনেকগুলোই দখবেন এক্স-উইনডোজ সংক্রান্ত, মানে, আমাদের এই পাঠমালার এলাকার বাইরে। ও, কী ভাগ্যি যে

তখন বাদ দিয়েছিলাম। অবশ্য আমি খুব একটা পারতামও না বুঝিয়ে বলতে, নিজেই তো আন্দাজে মান্দাজে করি। বেশিরভাগই। দু-একটা কাজ অবশ্য খুব ইন্টারেস্টিং লাগে বেশ, যেমন এক্সউইনডোজের কনফিগারেশন ফাইল 'XF86Config'-টা বদলানো। ফাইলটা থাকে '/etc/X11' ডিরেক্টরিতে। ন নম্বর দিনে আমাদের ফাইলসিস্টেম হায়েরার্কির আলোচনা থেকে মনে করে দেখুন তো ঠিকানাটা আমাদের দেওয়া ছকের সঙ্গে মিলিয়ে নিতে পারছেন কিনা। যাকগে, কেরমে কেরমে আমাদের শেষের শুরু চলে এল, এবারে ব্যাশ শেলের কনফিগারেশন ফাইল। এই সেকশনে আসতেই পারত, সেই ডট ফাইল বা আরসি ফাইল, ইচ্ছে করেই পরের সেকশনে নিয়ে গেলাম। শুধু একটা কথা, আগের দিন ভুলে গেছিলাম কনফিগারেশন ফাইলের বৃত্তান্ত লেখার সময় যে লেখাগুলো থেকে সাহায্য পেয়েছিলাম তাদের রেফারেন্স দিতে। সেইটা দিয়ে রাখি, আর মূল রেফারেন্স তো দেখেছেনই নিজের মেশিনের সুজে সিস্টেমের ডিরেক্টরিগুলো। এই বাইরে থেকে সাহায্যগুলো সবই নেট থেকে পাওয়া, মালগুলো আমার কাছে নামানোই আছে, যদি কারুর পড়তে ইচ্ছে হয়, আমারই মত যার নেট করার সময় পয়সা গুণতে হয়, মেল করুন আমায়, কিছু একটা করা যাবে। এই বইটা লিখতে লিখতে দিন শূন্য থেকে শুরু করে নেট থেকে পাওয়া যত লেখা কাজে লাগিয়েছি, সবই রাখা আছে আমার '/mnt/arkive/lesson.bkp' ডিরেক্টরিতে।

http://www.comptechdoc.org/os/linux/howlinuxworks/index.html
Linux Startup Scripts by kimo@debian.lib.monash.edu.au
http://www.comptechdoc.org/os/linux/usersguide/index.html
ftp://www6.software.ibm.com/software/developer/library/l-config.pdf

২।। ব্যাশ শেলের লোকাল কনফিগারেশন ফাইল

ব্যাশ শেলের ব্যাশ নামের ব্যাখ্যা তো আগেই এসেছে — বর্ন-এগেন-শেল (**B**ourne-**A**gain-**SH**ell)। বেল ল্যাবরেটরির ইউনিক্স রিসার্চ ভার্শন সেভেনের জন্যে বর্ন শেল লিখেছিলেন স্টিভ বর্ন। তার পর থেকে আরো অনেক শেল লেখা হয়েছে। কর্ন-শেল, সি-শেল ইত্যাদি। মূল বর্ন শেল যা যা পারত তা সবই পারে ব্যাশ, এবং আরো বেশি কিছু পারে। বর্ন শেলের জন্যে লেখা প্রোগ্রাম ব্যাশেও চালানো যায়। গ্নু-লিনাক্সে ডিফল্ট শেল এই ব্যাশ। যা সিস্টেম এবং ব্যবহারকারীর মধ্যে মাধ্যমের কাজ করে। ইউজারের কাছ থেকে আদেশ পেয়ে জেগে ওঠে ব্যাশ, পাঁচ নম্বর দিনের থেকে মনে করুন, এবং সেই আদেশ পাঠিয়ে দেয় কারনেলের কাছে, তারপর কাজ সমাপ্ত হলে তার ফলাফল ইউজারের কাছে পৌঁছে দিয়ে ফের ঘুমোতে চলে যায়। এর পরের সেকশনে শেলকে অনেকটা বিশদে বুঝতে হবে আমাদের, এখন আমরা এই ব্যাশ শেলের কনফিগারেশন ফাইলগুলো বোঝার চেষ্টা করব।

ব্যাশ কনফিগার করা হয় মূলত পাঁচটা ফাইল দিয়ে। '/etc/profile', '/etc/bashrc', '~/.bash_profile', '~/.bashrc' এবং '~/.bash_logout'। এর মধ্যে প্রথম দুটো গ্লোবাল কনফিগারেশন ফাইল, এদের কথায় আমরা আসছি এর পরের সেকশনে। পরের তিনটে হল একজন একক ইউজারের লোকাল ফাইল, এর সবকটা ফাইলই যে সবসময় থাকবে তা নয়, কিন্তু চাইলে এদের যে কাউকেই ব্যবহার করা যায়। যেমন, আমার মেশিনের সুজে সিস্টেমে আমার হোম ডিরেক্টরিতে সমস্ত ডটানন ফাইলের তালিকার মধ্যে ব্যাশ সংক্রান্ত ফাইল আছে '~/.bash_history', '~/.bashrc' এবং '~/.profile'। এই শেষটা মানে '~/.profile' হল ওই '~/.bash_profile' ফাইলেরই নামান্তর। আর প্রথম ফাইলটা, '~/.bash_history', সাত নম্বর দিনের ২.৩ সেকশন থেকে আমাদের চেনা, এখানে আমাদের আদেশের ইতিহাস থাকে, এটার কথায় আমরা পরে আসছি। কোনো '~/.bash_logout' ফাইল সুজেতে থাকেই না, তবে চাইলে কোনো ইউজার বানিয়ে নিতেই পারে। সুজেতে আমার হোম ডিরেক্টরি মানে '/home/dd' থেকে '.bashrc' আর '.profile' — এই ডটানন ফাইলদুটো তুলে দেওয়া যাক। খেয়াল করুন, এই ডিরেক্টরিটা কিন্তু ইউজার 'dd'-র বেলায় '~'। প্রথমে '/home/dd/.bashrc' —

```
# There are 3 different types of shells in bash:
# The login shell, normal shell and interactive shell.
# Login shells read ~/.profile and interactive shells read ~/.bashrc;
test -s ~/.alias && . ~/.alias
if [ -x /usr/bin/fortune ] ; then
    echo -e '\n***\n'>>F;/usr/bin/fortune -os | tee -a F | cat
fi
```

এর শেষ চারটে লাইন আপাতত খ্যামা দিন, পরে আসছি, শুধু প্রথম চারটে লাইন, কমেন্ট লাইন, '#' লাগানো, আপাতত আপনার হজমনীয়। দেখুন, এইমাত্র যা বললাম, তার সঙ্গে মেলাতে পারছেন? এবার আসুন '/home/dd/.profile' পড়া যাক —

```
# This file is read each time a login shell is started.
# All other interactive shells will only read .bashrc;
test -z "$PROFILEREAD" && . /etc/profile
if [ -x /usr/bin/fortune ] ; then
    echo -e '\n***\n'>>F;/usr/bin/fortune -s | tee -a F | cat
fi
```

এখানেও আপাতত শুধু প্রথম দুটো লাইন পড়ুন। শুধু একটা মজা দেখুন, শেষ তিনটে লাইন দুটো ফাইলেই হুবহু এক — স্বাভাবিক, ওটা আমার অবদান, আর যা হয়, নিজের অবদান মাত্র একবার ঘোষণা করে সুখ কই? শুধু ফরচুনের অপশনে, '.bashrc' ফাইলে আছে '`/usr/bin/fortune -os`', আর '.profile' ফাইলে আছে '`-s`'। এটার মানে আর কিছু নয়, এইমাত্র কমেন্ট লাইনে যা পড়লেন, লগ-ইনের সময়ে '.profile' পড়ে ব্যাশ শুধু একটা ছোট ফরচুন শোনায়, '`s`' মানে শর্ট বা ছোট। আর আমি যখন কোনো জ্যান্ত শেলকে ডেকে আনি, শেলে কোনো কাজ করার জন্যে, যাকে ব্যাশের ম্যানপেজের ভাষায় বলে ইন্টারাক্টিভ ইনভোকিং, এক্স-উইনডোজে থাকাকালীন কোনো টার্মিনাল খোলা মানেও তাই, তখন ব্যাশ পড়ে নেয় '.bashrc', তখন শোনায় ওই আপত্তিকর বাণী, '`o`' মানে অফেনসিভ। একটু পরে ওই বাদ দেওয়া লাইনগুলোর মানে আর এই জায়গাগুলোও আরো স্পষ্ট হয়ে উঠবে। কিন্তু এর মানে এই নয় যে '~/.bashrc' বা '~/.profile' ফাইলে শুধু এই-ই থাকবে, এছাড়া আরো বহু কিছু থাকতে পারে এবং থাকেও, ইউজারের ইচ্ছা অনুযায়ী। কী কী ভাবে কী আনা যায় সেখানে সেগুলোর আলোচনায় আমরা আসছি। আমি এখানে এদুটোকে তুলে জাস্ট আপনাদের একটু অভ্যস্ত করলাম, আর এদের মধ্যে রাখা ওই কমেন্ট লাইনগুলো পড়াতেও চাইছিলাম। সুজে সিস্টেমে কোনো '~/.bash_logout' নেই, চাইলেই রাখা যেত। কিন্তু আমার মেশিনের অন্য সিস্টেমের ইউজার 'dd'-র হোম ডিরেক্টরি থেকে '.bash_logout' ফাইলটা তুলে দিই —

```
# ~/.bash_logout
clear
```

এখানে দেখুন, আর কিছুই নেই, কিন্তু চাইলেই রাখা যেত। এই '~/.bash_logout' হল একজন ইউজার লগ-আউট করে যাওয়ার পূর্ব-মুহূর্তে ঠিক কী কী কাজ সিস্টেমকে করে নিতে হবে, তার তালিকা। যেহেতু এটা এই ইউজারের লোকাল, তাই একজনের '~/.bash_logout' ফাইলে অন্য ইউজারের কিছু এসে যায়না।

'~/.bash_profile' বা '~/.profile' ফাইলে থাকে একজন একক ইউজারের নিজস্ব ব্যাশ কনফিগারেশন। এই ইউজার যখন ব্যাশ চালায় তখন তার এনভায়রনমেন্ট ভ্যারিয়েবলগুলোকে নিয়ন্ত্রণ করে এই '~/.bash_profile'। গ্লোবাল সেটিং-এ '/etc/profile'-এর যে ভূমিকা, লোকাল সেটিং-এ সেই একই ভূমিকা '~/.profile' ফাইলের। '/etc/profile' নিয়ন্ত্রণ করে সমস্ত ব্যবহারকারীর ব্যাশকে, আর '~/.profile' নিয়ন্ত্রণ করে একজন ব্যক্তি ব্যবহারকারীর ব্যাশকে। '~/.bashrc' ফাইলও তেমনি নিয়ন্ত্রণ করে একজন ব্যক্তি ব্যবহারকারীর নিজের পছন্দমত অ্যালিয়াসগুলোকে এবং নিজের পছন্দমত বিভিন্ন ফাংশনের ব্যবহারকে। এই অ্যালিয়াসের কথায় আমরা একটু বাদেই আসছি।

৩।। ব্যাশের গ্লোবাল কনফিগারেশন

আমরা যে পাঁচটা ফাইলকে ব্যাশের কনফিগারেশন বলে উল্লেখ করলাম, '/etc/profile', '/etc/bashrc', '~/.bash_profile' (বা '~/.profile'), '~/.bashrc', এবং '~/.bash_login', তার মধ্যে দেখুন, প্রথম দুটো, '/etc' ডিরেক্টরিতে — এরা হল গ্লোবাল কনফিগারেশন। মানে, এই সিস্টেমে যারা যারা ব্যাশ ব্যবহার করে তাদের সবারই ব্যাশকে নিয়ন্ত্রণ করবে এই ফাইলদুটো। যেমন আমরা আগেই বলেছি, গ্লোবাল কনফিগারেশন, তাই এদের সাধারণত এই ডিরেক্টরিতেই পাওয়া যাওয়ার কথা। পরের তিনটে হল লোকাল কনফিগারেশন, মানে একজন ব্যক্তি ইউজারের নিজের ব্যাশ-ব্যাবহারকে কনফিগার করা যায় এদের দিয়ে। নিজের হোম ডিরেক্টরিতেই গোপন বা হিড্ন ফাইল হিশেবেই থাকে এরা, কারণ এরা ডটানন, মানে মুখে ডট আছে এদের। স্বাভাবিক রকমে '`ls`' করলে যাদের দেখা যায়না, '`ls -a`' কমান্ড দিলে, মানে অল বা সব ফাইল দেখাতে বললে তখন এদের দেখা যায়। আর যেমন বললাম,

যদি এদের সব ফাইলকে আপনি আপনার হোম ডিরেক্টরিতে না পান, কুছ পরোয়া নেই, নিজের সিন্ধু দর্শনের বিন্দু ফাইল নিজেই বানান।

আর যে কোনো সময়েই আপনার করা বদলে ঘেঁটে যাওয়া ব্যাশ ঠিক করে নেওয়ার জন্যে আপনার সিস্টেমে '/etc/profile' তো আছেই — ব্যাশের গ্লোবাল কনফিগারেশন। সাত নম্বর দিনে যে এনভায়রনমেন্ট ভ্যারিয়েবল বা গেরস্থালির চলরাশির কথা উল্লেখ করেছিলাম, মনে করে দেখুন, '`set`' কমান্ড থেকে পাওয়া ফলাফলকে '`less`' করে পড়ে পড়ে — সেই চলরাশিগুলোকে নিয়ন্ত্রণ করে এই '/etc/profile'। ব্যাশ থেকে ব্যাশের উপর দাঁড়িয়ে যে প্রোগ্রামগুলো চালাতে হয়, ব্যাশস্ক্রিপ্টগুলো যেমন, সেগুলোর চলা এবং কাজ করাকেও নিয়ন্ত্রণ করে। মানে, যে কোনো ইউজার যখনি কোনো ব্যাশ প্রোগ্রাম বা স্ক্রিপ্ট চালায় তার ঘর-গেরস্থালির আকার-আকৃতি, মানে এনভায়রনমেন্ট ভ্যারিয়েবলদের মান কী হবে সেটা বলে দেয়। আপনাদের মধ্যে যারা উইনডোজ এমিগ্রি, মানে উইনডোজ জগত থেকে গ্নু-লিনাক্সে অভিবাসী, তারা মনে করতে পারবেন, 'c:\autoexec.bat' ফাইল দিয়ে যে কাজগুলো করতেন, প্রম্পটের আকার বদলানো, কোন কোন ডিরেক্টরি কাজ করার পথনির্দেশে থাকবে, ইত্যাদি, তাদের সবগুলো কাজই করা যায় এই '/etc/profile' ফাইল দিয়ে, এবং, স্বাভাবিক ভাবেই, আরো অনেক বেশি কিছু করা যায়। এরকমই আর একটা ফাইল '/etc/bashrc' — ব্যাশের আর একটা গ্লোবাল কনফিগারেশন ফাইল। কী কী ওরফ বা অ্যালিয়াস থাকবে আপনার, বা ব্যাশ চালু হওয়ার সময় কী কী ফাংশন বা কমান্ড বা স্ক্রিপ্ট গোড়াতেই চালিয়ে বা এক্সিকিউট করে নেবে, তার হদিশগুলো থাকে এই '/etc/bashrc' ফাইলে। আর অ্যালিয়াস মানে শর্টকাট, অল্প কথায় ব্যাশকে বড় বড় আদেশ দিতে পারা। একটু দাঁড়ান, পরেও লাগবে, তাই আমরা একটু অ্যালিয়াস নিয়ে দু-চার কথা বলে নেব এর পরের সেকশনে। তারপরে আবার ফিরে আসব ব্যাশের কথায়। অ্যালিয়াস ছাড়া এই গ্লোবাল কনফিগারেশন ফাইল '/etc/bashrc'-এ আর যা থাকে তা হল ব্যাশ চালু হওয়ার সময় যে যে ফাংশন চালিয়ে নেওয়া তাদের তালিকা। এই ফাংশনগুলো সবই একটা একটা ছোট ছোট স্ক্রিপ্ট, ব্যাশ সঠিক জায়গায় যাদের দরকারমাফিক চালিয়ে নেয়। যদি একটা সিস্টেমে '/etc/bashrc' ফাইল না থাকে, তাহলে এই সমস্ত জিনিষ দেওয়া থাকে '/etc/profile' ফাইলেই। কখনো কখনো ফাইলের নামটা একটু ভিন্ন হতে পারে, যেমন আমার সুজে সিস্টেমে ফাইলটার নাম '/etc/bash.bashrc'। সেখানে '/etc/profile' ফাইলের সাইজ ২৮৫ লাইন আর '/etc/bash.bashrc' ফাইলের সাইজ ২৫৩ লাইন।

৪।। অ্যালিয়াস বা ওরফ

এই অ্যালিয়াস বা ওরফ মানে ফাটা-রতন বা কাটা-গমা বললেই পুলিশ যেমন চিনে যায় এর পিছনে পিতৃদত্ত নামে কমল চন্দ্র সাহা বা পরেশ মাইতি, মানে এলাকার কোনো স্বনামধন্য SমাজSেবীকে, এবং এদের কেউ কোনো নতুনতর সমাজসেবায় হাত দেওয়া মাত্রই নিট সেই মস্তানের কাছে তোলার রেট বাড়ায়। কারণ, ওই ওরফ বা অ্যালিয়াসটার পিছনে মূল ঘুঘুটাকে সবচেয়ে ভালোভাবে চেনে পুলিশ, তার পিছনে তার দাদা কাম পিতাকেও, তার আমদানির কারণেই চিনে রাখতে হয়। তেমনি, ঠিক আমদানি নয়, কমান্ড প্রম্পটে আপনার কাজকর্মের সুবিধের জন্যে সিস্টেমে আপনি কিছু ওরফ বা অ্যালিয়াস বানিয়ে নিতে পারেন। ডিস্ট্রোর তরফ থেকে নিজেরই বানানো ফাইল থাকে একটা। দাঁড়ান, উদাহরণ দিয়ে বোঝানো যাক। আপনার মেশিনে মোট কী কী অ্যালিয়াস ডিস্ট্রো থেকেই করা আছে সেটা জানতে হলে আপনাকে অ্যালিয়াসের ম্যানপেজ পড়তে হবে। কিন্তু অ্যালিয়াসের কোনো নিজের ম্যানপেজ নেই। অ্যালিয়াস হল ব্যাশের একটা আভ্যন্তরীন কাজ, বিল্ট-ইন ফাংশন। ব্যাশের ম্যানপেজ পাবেন ম্যানের সেকশন একে। আগের দিন আমরা যে ম্যানুয়ালের ওয়েবপেজগুলো বানিয়েছি তার থেকে 'bash.1.html'-টা খুলুন আপনার ব্রাউজার দিয়ে। অনেকটা পরের দিকে দেখুন একটা গোটা সেকশনই আছে 'SHELL BUILTIN COMMANDS' নামে। তার ভিতরেই পাবেন 'alias'-এর বিবরণ। দেখুন, দেওয়া আছে, কমান্ড প্রম্পটে শুধু '`alias`' বা '`alias -p`' কমান্ড আপনাকে আপনার সিস্টেমে গোটা অ্যালিয়াসের তালিকাটা দেখাবে। এবার আমার মেশিনের সুজে সিস্টেমের এই তালিকাটাকে রিডাইরেক্ট করে একটা ফাইলে এনে তার থেকে কয়েকটা লাইন তুলে দিই।

```
alias l='ls -alF'
alias la='ls -la'
alias ll='ls -l'
alias ls-l='ls -l'
```

এখানে শুধু 'ls' সংক্রান্ত কয়েকটা অ্যালিয়াস তুলে দিলাম, আছে কিন্তু প্রচুর। দেখুন, পড়ুন, মনে রাখুন। এবার এর মধ্যে আমাদের খুব পরিচিত হল দু-নম্বরটা। লিংক বুঝতে গিয়ে আমরা বারবার '`ls -l`' ব্যবহার করেছি। তার সঙ্গে আর একটা অপশন যোগ হয়েছে, '`a`'। সেটাও আমাদের চেনা, অল বা সমস্ত ফাইল দেখানোর, গোপন বা হিড্ন গুলোকেও। এবার দেখুন, অ্যালিয়াসটা তৈরি করা আছে বলে আমি কমান্ড প্রম্পটে '`la`' লিখে এন্টার মারলেই ও আসলে যে আদেশটা পালন করবে সেটা হল '`ls -la`'। কারণ, '/etc/bashrc' ফাইলে সেইরকমই বলে দেওয়া আছে।

এরকম আপনি নিজেও বানিয়ে নিতে পারেন। ধরুন আপনি চাইছেন, আপনি আপনার নতুন প্রবন্ধে হাত দেওয়ামাত্র ব্যাকগ্রাউন্ডে গান বাজতে শুরু করবে। গান না-শুনে লেখা যায়? আর গানগুলো আছে আপনার হোম ডিরেক্টরির ভিতর, 'music' বলে একটা ডিরেক্টরিতে, আর এগুলো সব 'mp3' ফাইল। 'mp3' কাকে বলে যদি না-জানেন তো বলে রাখি, সাধারণ অডিও মানে শব্দতথ্যের কোঁকড়ানো ছোটকরা একটা আকার। কতটা ছোট সেটার একটা আন্দাজ পাবেন নাম থেকে, এক একটার ভিতর তিনপিস করে ভুঁড়িদার হাতিমি সাইজের এমপি মানে মেম্বার অফ পার্লামেন্ট ধরে যায় — এমপি-র সংখ্যা তিন বা থ্রি। নামের এই বিশ্লেষণ যদি আপনার পছন্দ না-হয়, অন্য বিশ্লেষণটা কী সেটা নিজে খুঁজে নিন। আপনার সিস্টেমের হাউ-টু গুলোর ভিতরেই পেয়ে যাবেন।

এবং আপনি আপনার প্রবন্ধগুলো সব রেখেছেন আপনার হোম ডিরেক্টরির ভিতর 'probondho' ডিরেক্টরিতে। এবার আপনি এরকম নতুন একটা ওরফ বা অ্যালিয়াস বানাতে চান যা আপনার প্রবন্ধ লেখার ডিরেক্টরিতে নতুন প্রবন্ধ শুরু করার জন্যে আপনাকে ইম্যাক্স খুলে দেবে, এবং একই সঙ্গে ব্যাকগ্রাউন্ডে গান বাজাতে শুরু করবে, তাহলে কমান্ড দিন, '`alias likhi='mpg123 ~/music/*.mp3 & cd ~/probondho;emacs'`' এবং এন্টার মারুন। আর একবার '`alias`' কমান্ডটা মেরে দেখে নিন, এবার আগেরগুলোর সঙ্গে ও আপনার এই নতুন বানানো অ্যালিয়াসটাও যোগ করে দিয়েছে। এবার থেকে '`likhi`' কমান্ড দিলেই ব্যাশ আপনাকে এনে দেবে '~/probondho' ডিরেক্টরিতে এবং ইম্যাক্স খুলে দেবে আপনার জন্যে, আর ব্যাকগ্রাউন্ডে বাজতে থাকবে আপনার '~/music' ডিরেক্টরির '*.mp3' ফাইলগুলো একের পর এক। গানটা কিন্তু সত্যিই ব্যাকগ্রাউন্ডে চলে গেছে। আমাদের দেওয়া আদেশের মধ্যে '`&`' চিহ্নটা থাকায় ব্যাশ এটা করেছে। লিখতে লিখতে গানটা যদি কখনো বন্ধ করতে চান, তাহলে কিন্তু সরাসরি হবেনা, প্রথমে আপনাকে কাজ করতে থাকা ইম্যাক্সটাকে ব্যাকগ্রাউন্ডে নিয়ে যেতে হবে, '<Ctrl><Z>' সুইচ টিপে। দেখবেন ইম্যাক্সটা হাওয়া হয়ে গিয়ে কমান্ড প্রম্পট ফেরত এসেছে, কিন্তু ইম্যাক্স তাই বলে আদিউ জানিয়ে মহাপ্রস্থানে চলে গেছে তা কিন্তু নয়, জাস্ট ওয়াভোয়া বলেছে, এখুনি একবার '`fg`' লিখে এন্টার মারলেই ফেরত চলে আসবে স্ক্রিনে। ইম্যাক্সটা আসলে ব্যাকগ্রাউন্ডে চলে গেছে, বাজতে থাকা গানের ওই 'mpg123' প্রোগ্রামের মত। ইম্যাক্সকে যেভাবে একবার স্রেফ '`fg`' সেধেই ফেরত আনতে পারলাম আমরা, 'mpg123' প্রোগ্রামটা কিন্তু তা হবেনা। একেও ফোরগ্রাউন্ডে আনতে হবে '`fg`' মেরেই। কিন্তু, এখানে একটু মৃদু জাটিল্য আছে। এখন কিন্তু ব্যাকগ্রাউন্ডে একটা নয়, দুটো জব বা কাজকে আপনি পাঠিয়েছেন। তাহলে? '`fg`' মারলে কে উঠে আসবে ফোরগ্রাউন্ডে? জব এক না জব দুই? আসবে শেষ যাকে আপনি ব্যাকগ্রাউন্ডে পাঠিয়েছিলেন, মানে এই উদাহরণের ইম্যাক্স। তাহলে, গান থামাবেন কী করে? 'পায়েলিয়া গান থামা এবার' গাইবেন? বেগম আখতারের মত গাইতে পারলে সিস্টেম কখনো কখনো রিঅ্যাক্ট করে বলেই শুনেছি, কিন্তু আমরা যারা 'সুরের গুরু দাও গো সুরের দীক্ষা' গাইলেই অহল্যা জেগে ওঠে পাষাণ ফুঁড়ে, ট্রাফিক থেমে যায়, এবং নিকটতম অ্যামওয়েজ বিক্রেতা নাগা-সন্ন্যাসের সংকল্পে হিমালয়ের টিকিট কাটে, তাদের কী হবে?

তাদের জন্যে আছে জবস। কমান্ড দিন '`jobs`'। দেখুন, একটা তালিকা ফুটে উঠেছে। ফল্গুর বালির তলায় অন্তঃশীলা জলের প্রবাহের মত আপাতঅদৃশ্য কাজের মিছিল, হায় বাংলার অর্থনীতিতেও যদি একবার জবস মারা যেত। প্রত্যেকটা জবের নাম পাশে একটা করে সংখ্যা। যদি আপনি আর কোনো কাজকে ইতিমধ্যে, মানে শেষবার লগ-ইন করার পর থেকে আর ব্যাকগ্রাউন্ডে না-পাঠিয়ে থাকেন, তাহলে দেখুন, তালিকায় স্পষ্ট দেখাচ্ছে আপনার ওই 'likhi' কমান্ড থেকে তৈরি হওয়া নেপথ্যসঙ্গীত 'mpg123' আর আপনার নিজের হাতে অন্দরে পাঠিয়ে দেওয়া 'emacs', পর পর, তাদের নম্বর দুই আর এক। ইম্যাক্স এক কারণ শেষ সক্রিয় ছিল সে, এবং তার আগে সক্রিয় ছিল 'mpg123', তাই তার নম্বর দুই। জবসের এরকম একটা সম্ভাব্য বাণী এখানে একটু বুঝে নেওয়া যাক।

```
[1]-  Running        mpg123 ~/music/*.mp3 &  (wd: ~/probondho)
[2]+  Stopped        emacs
```

এবার আপনি ‘fg 1’ কমান্ড দিলে ফোরগ্রাউন্ডে আসবে ইম্যাক্স, এবং ‘fg 2’ কমান্ড দিলে ফোরগ্রাউন্ডে আসবে ‘mpg123’। এদের মধ্যে ইম্যাক্সকে টাটা জানানোর উপায় তো জানেনেই, ‘<Ctrl><X><Ctrl><C>’। আর ‘mpg123’-কে টাটা জানানোর উপায় হল ‘<Ctrl><C>’। তবে যেহেতু এখানে একাধিক গান পরপর কিউতে বা সারিতে দেওয়া আছে ‘~/music/*.mp3’ করে, তার মানে ‘~/music’ ডিরেক্টরিতে মোট যত ‘*.mp3’ ফাইল আছে তাদের চালানোর আদেশ, তাই একবার ‘<Ctrl><C>’ মারলে ব্যাশ মনে করবে আপনি পরের গানটা শুনতে চাইছেন। না-থেমে, পরপর দুবার ‘<Ctrl><C>’ মারলে ও বেরিয়ে যাবে ‘mpg123’ থেকে। আরো উপায় থাকতে পারে, আমার ভালো মনে পড়ছে না, ‘man mpg123’ বা ‘mpg123 --help’ করে দেখে নিন।

জবস কমান্ডের যে বাণীটা আমরা তুলে দিলাম, তার দু-একটা বিষয় এখনো আমাদের খেয়াল করার আছে। যেমন দেখুন উপরের লাইনে একদম ডানদিকে ব্রাকেটের ভিতর ‘wd’ — কোন শব্দবন্ধকে ছোট করে এটা করা হয়েছে বুঝতে পারছেন কি? ‘pwd’ কমান্ডটা মনে আছে? বারবার ব্যবহার করেছি আমরা, সেটাও যদি মনে না-থাকে ম্যানুয়াল থেকে দেখুন, তারপরেও মনে না-পড়লে, একটা টোটকা বলে দিই, খুব কাজে লেগে যায়, যে কোনো ভুলে যাওয়া কথা মনে পড়ানোয় — নিজের বাঁদিকের কানটা নিজে কামড়ে দেখুন, তবে কানটা খুলে আনবেন না, শুধু দাঁত বাড়িয়ে কামড়াবেন। তবে, কানে কোনো পুরোনো ইনফেকশন থাকলে কিন্তু করতে নেই। আর একদম বাঁদিকে, দুটো লাইনেই, চৌকো ব্রাকেটের ডান-পাশেই একটা ‘-’ আর একটা ‘+’ চিহ্ন। এই দুটোর মানে কী, সেটাও আমি বলে দেবনা, শুধু বলতে পারি যেরকম ফোরগ্রাউন্ড বা ‘fg’ কমান্ডের সঙ্গে একটা এক বা দুই দিয়ে আমরা একটা চলমান কিন্তু আপাতঅদৃশ্য জবকে পুরোভূমিতে বা ফোরগ্রাউন্ডে আনছিলাম, সেই রকম এই দুটো চিহ্ন দিয়েও করা যায়। এর গোটাটাই এবং আরো অনেককিছু পাবেন ‘bash’-এর ম্যানুয়াল পেজে, দেখুন ‘Job control’ নামে একটা আলাদা গোটা সেকশনই আছে ব্যাশের ম্যানুয়ালে। অ্যালিয়াস (alias) বা ওরফ যেমন ব্যাশ শেলের একটা অন্তর্বর্তী আদেশ বা শেল বিল্ট-ইন কমান্ড, সেরকম জবস (jobs) বা ক্রিয়াশীল কাজের তালিকা দেখানোটাও ব্যাশের ভিতরকার কাজগুলোর একটা।

এইমাত্র যে অ্যালিয়াসটা আপনি বানালেন সসঙ্গীত সাহিত্যচর্চার, সেটা কিন্তু অস্থায়ী অ্যালিয়াস। পুলিশের রেকর্ডে এখনো তার নাম তোলা হয়নি। পরেরবার লগ-ইন করে আপনি আর পাবেন না অ্যালিয়াসটা। তাহলে কী করবেন, পরের বার পেতে হলে? কোনো একটা কনফিগারেশন ফাইলে যোগ করে দেবেন। গ্লোবাল নয়, লোকাল ফাইল। আপনার শিল্পমনন অন্যদের উপর চাপানোর কোনো অধিকার আপনার নেই। কিন্তু নিজের জন্যে এটাকে তুলে রাখতেই পারেন, আপনার হোম ডিরেক্টরির ডটানন ফাইল ‘~/.bashrc’ ইম্যাক্স দিয়ে খুলে তার একদম শেষে হুবহু একটু আগের আমাদের অ্যালিয়াস বানানোর গোটা লাইনটা টাইপ করে তুলে দিন। এমনকি যদি ফাইলটা এমনিতে আপনার হোমে নাও থাকে, একটু আলতো করে ছুঁয়ে দিন, টাচ কমান্ডটা তো মনে আছে, ‘touch ~/.bashrc’। তারপর ইম্যাক্স দিয়ে খুলে টাইপ করে দিন। বা, সরাসরি ‘cat>>~/.bashrc’ কমান্ড দিয়ে কমান্ড প্রম্পটে গোটা লাইনটা টাইপ করে দিয়ে, ‘<Ctrl><D>’ মেরে ক্যাট করা শেষ করুন। হিল্লে হয়ে গেল। এই ‘>>’ অপারেটরটা মনে পড়ছে তো, শির জিনিষ, যদি ফাইল থাকে তো তার শেষে লাগিয়ে দেবে, আর না-থাকে তো বানিয়ে লাগিয়ে দেবে। এতে সুবিধেটা এই যে ওই নামে ফাইলটা থাকলেও সেটাকে উড়িয়ে দেবেনা, এমনিতে ক্যাট যা করে। এর পর থেকে প্রত্যেকবার লগ-ইন করার পর এই অ্যালিয়াসটা আপনা থেকেই পাবেন। তবে, ‘~/.bashrc’ ফাইল এমনিতেই আপনার হোমে থাকলে, ডিফল্ট ব্যবস্থায় সেটাই থাকার কথা, একটা নিরাপত্তা ব্যবস্থা আগেই করে রাখুন, কমান্ড দিন, ‘cp ~/.bashrc ~/.bashrc.bkp’। এতে সুবিধেটা এই যে, নিজে করতে গিয়ে ঘেঁটে ফেললে, পরে ফেরত পেয়ে যাবেন আদত ‘.bashrc’ ফাইলটা ‘.bashrc.bkp’ নামে। তখন পুরোনোটা ফেরত পেতে গেলে জাস্ট একটা কমান্ড দিন, ‘mv ~/.bashrc.bkp .bashrc’, মানে পুর্নব্যাশো-ভব করে দিলেন। কনফিগারেশন ফাইল নিয়ে নখরাবাজি করার আগে এই কবচকুণ্ডলটা সবসময় মনে রাখবেন। কারন, শিখতে গেলে সিস্টেম ঘাঁটবেই, সেই প্রাচীন প্রবাদটা তো শুনেছেন নিশ্চয়ই — “কঁপিতে শেখেনা কেহ না-ঘেঁটে মেশিন” — ‘কঁপিতে’ শব্দটা হল ‘কম্পিউটার-করা’ নামক ক্রিয়াপদের প্রাচীন মৈথিলী ব্রজবুলি আকার, সুনীতি চাটুজ্জের ওডিবিএল-এ দেওয়া আছে।

৫।। ব্যাশ এবং তার ভ্যারিয়েবল

একটু আগে ব্যাশের কনফিগারেশন ফাইলগুলো আলোচনা করতে গিয়ে বারবার এই এনভায়রনমেন্ট ভ্যারিয়েবলের কথা আসছিল। সাত নম্বর দিনে এই এনভায়রনমেন্ট ভ্যারিয়েবল বা প্রতিবেশ-চল কাকে বলে তা আমরা একটু ছুঁয়ে চলে এসেছিলাম, আজ আমরা দেখার চেষ্টা করব, কোথা সে পথের শেষ, ওহে চঞ্চল। চঞ্চল, মানে চলে বেড়ায়, মানে ভ্যারি করে বা ভ্যারিয়েবল, সেই জন্যেই এদের বলে চল বা চলরাশি। রাশিটা আসে গণিতে ব্যবহারের সূত্রে। শূন্য এক আর দুই নম্বর দিনে আমরা বারবার স্মৃতিভূমির কথা এনেছি, নানা ধরনের নানা আকারের স্মৃতিভূমি। কখনো তা হার্ডডিস্কের মত স্মৃতির ভাঁড়ার, কখনো র‍্যামের মত উদ্বায়ী স্মৃতি। পরে আট নম্বর দিনে এই র‍্যামকে দিয়ে গোটা কাজ করতে গিয়ে অপারেটিং সিস্টেমের কারনেলের কাছে যাতে স্মৃতি নামক রসদের যাতে টান না-পড়ে, সেই জন্যে, ১২৮ বা ২৫৬ বা ৫১২ যত এমবি ভৌত র‍্যামই থাক, তার সঙ্গে সঙ্গে পোঁ-ধরার মত একটা ভৌতিক স্মৃতি বা ভারচুয়াল মেমরির কথা বলেছি। সোয়াপ ফাইলের ফাইল সিস্টেমের কথা বলতে গিয়ে এবং হার্ডডিস্কের বিভিন্ন ধরনের ব্লকের আলোচনা করতে গিয়ে আমরা বলেছি, কী ভাবে একটা গ্নু-লিনাক্স সিস্টেম তার প্রয়োজনীয় সমস্ত কিছুকে এই ভারচুয়াল মেমরিতে তুলে নেয়, এবং লগ-আউট বা শাট-ডাউনের সময় সেই গোটা ভৌতিক ক্রিয়াকলাপটাকে ফ্লাশ করে, মানে ভৌত ডিস্কে লিখে দিয়ে বেরিয়ে আসে।

আমরা যখন পাঁচ নম্বর দিনে, ‘dd’ নামের ইউজার হয়ে ‘`echo $PATH`’ মেরে একক ব্যক্তি ইউজার কোনো কমান্ড দিলে ব্যাশ শেল কোথায় কোথায় ওই কমান্ডে দেওয়া নামের প্রোগ্রাম ফাইলটা খুঁজবে তার তালিকা, মানে পথনির্দেশ দেখেছিলাম, বা, ‘root’ হয়ে মানে সুপার-ইউজার হয়ে ওই একই কমান্ড দিয়ে রুট ইউজারের শেলের পথনির্দেশ দেখেছিলাম, তখন ব্যাশ আসলে কী করছিল? যে ইউজারের হয়ে আমরা কমান্ডটা চালাচ্ছি, তার জন্যে, সিস্টেমের স্মৃতিভূমিতে সে ‘PATH’ নামের এক ফালি জমি তুলে রেখেছে। সেই জমিতে যা ফলে আছে সেই ফলনটা ইকো বা প্রতিধবনি করে দিচ্ছিল স্ক্রিনে। এই জমিটা হল ‘PATH’ নামের একটা ভ্যারিয়েবল। আর ফলনটা হল তার মান বা ভ্যালু, আমরা তো ইতিমধ্যেই জানি ‘$’ দিয়ে ভ্যালুকে বোঝানো হয়। তাই ‘$PATH’ হল ‘PATH’ নামক ভ্যারিয়েবলের ভ্যালু। এই দুটো পথনির্দেশ যে আলাদা হয় তাও আমরা জানি। দেখেছি বারবার। এবং এখন এটাও আমরা জানি রুটের খেতের তরমুজ কী করে এলেবেলে ইউজারের খেতে করমচা হয়ে যাচ্ছে। যে কোনো ইউজার যেই লগ-ইন করছে, সে এমনি ইউজার হোক, বা রুট হোক, বা এমনি ইউজার ‘`su`’ করে রুট হোক, তার প্রত্যেকবারই একটা করে আলাদা আলাদা ব্যাশ শেল চালু হচ্ছে। দুই নম্বর দিনের বিভিন্ন ধরনের মাল্টিপ্লেক্সিং-এর আলোচনা মনে করুন, একই সঙ্গে চলমান রাশি রাশি প্রক্রিয়া বা প্রসেসের সঙ্গে একটা আদিম চিড়িয়াখানার সেই তুলনা। সেই প্রসঙ্গে বারবার এসেছে প্রসেসের বা প্রক্রিয়ার কথা — একটা প্রোগ্রাম ফাইলকে গ্নু-লিনাক্সে সরাসরি চালানো হয়না, তার একটা প্রতিরূপকে তুলে নেওয়া হয় ভৌতিক স্মৃতিতে। সেটাই একটা প্রসেস। ঠিক সেরকম, যখনই কোনো ইউজার লগ-ইন করে, বা কোনো ইউজার ইন্টারাক্টিভলি মানে জ্যান্ত পারস্পরিক রকমে ব্যাশকে চালায়, তখনই সিস্টেম ব্যাশ নামক প্রোগ্রাম ফাইলের একটা প্রতিরূপ নিজের স্মৃতিতে তুলে নেয়, জন্ম নেয় আর একটা প্রসেস বা প্রক্রিয়া। খুব সুন্দর করে এই প্রসেসের ব্যাপারটা বোঝানো আছে ট্যানেনবমের ‘মডার্ন অপারেটিং সিস্টেমস’ বইটাতে, যার রেফারেন্স আগেও একাধিকবার দিয়েছি আমরা। আর একান্তভাবে গ্নু-লিনাক্সের বেলায় এই ব্যাপারটা প্রাথমিকভাবে বোঝার জন্যে আছে এরিক রেমন্ডস-এর ‘ইউনিক্স অ্যান্ড ইন্টারনেট ফান্ডামেন্টালস হাউ-টু’, হাউ-টু গুলোর মধ্যেই পাবেন। প্রয়োজনীয় সমস্ত খুঁটিনাটিই সিস্টেমের ভিতরেই আছে, নিজে খুঁজে নিন।

যখনই একটা নতুন ব্যাশ-প্রক্রিয়া চালু হচ্ছে, সে এমনি ইউজারের জন্যে হোক, বা রুট ইউজারের জন্যেই হোক, ব্যাশ চালু হওয়ার আগে ওই কনফিগারেশন ফাইলগুলো পড়ে নিচ্ছে, এবং সেইমত প্রতিটা এনভায়রনমেন্ট ভ্যারিয়েবলের জন্যে একটা করে স্মৃতিখণ্ড বরাদ্দ করছে, আর যখনই কোনো প্রয়োজন পড়ছে, যেমন আমরা যখন পথনির্দেশ জানতে চাইলাম, ও সেখান থেকে পড়ে নিচ্ছে সেই প্রতিবেশ-চলর মান এবং তাকে ফুটিয়ে তুলছে স্ক্রিনে। ব্যাশে কিন্তু শুধু প্রতিবেশ-চল না, আরো এক রকমের চল বা ভ্যারিয়েবল হয়, তাকে বলে লোকাল বা আঞ্চলিক ভ্যারিয়েবল। এই আঞ্চলিক বা লোকাল ভ্যারিয়েবলগুলোকে বানিয়ে তোলা হয় একজন ব্যক্তি ইউজারের ইচ্ছা অনুযায়ী। বানিয়ে তোলাটা ঘটতে পারে দুই ভাবে। এক, ব্যক্তি ইউজার সরাসরি বানাল, কমান্ড প্রম্পটে কমান্ড দিয়ে। দুই, ব্যাশ চালু

হওয়ার সময় যে লোকাল কনফিগারেশন ফাইল পড়ে নেয়, ব্যক্তি ইউজারের হোম ডিরেক্টরি থেকে, সেই ফাইলগুলো মারফত।

সরাসরি আপনি যখনি ইন্টারাকটিভলি মানে 'bash' কমান্ড দিয়ে কোনো ব্যাশ চালু করলেন, এবং তার মধ্যে কোনো একটা ভ্যারিয়েবল বানিয়ে গুঁজে দিলেন, এই ব্যাশটা, মানে এই ব্যাশের এই চলমান প্রতিরূপ প্রক্রিয়ায় সেই ভ্যারিয়েবলের নামে একটা জমি তৈরি হল, এবং সেখানে তার মানটা গুঁজে দেওয়া হল। কিন্তু এই 'exit' কমান্ড দিয়ে বা '<Ctrl><D>' করে ব্যাশ থেকে বেরিয়ে যাওয়া মানে এই ভ্যারিয়েবলটারও ইন্তেকাল হয়ে গেল। এবার, একটা জিনিষ খেয়াল করুন। বেরিয়ে তো গেলেন এই ব্যাশ থেকে, কিন্তু বেরিয়ে গেলেন কোন চুলোয়? কেন? ব্যাশের বাইরে? না, সরি, সেটা যাওয়ার উপায় নেই, পাঁচ নম্বর দিনের আলোচনাটা মনে করুন, কোনো ইউজার লগ-ইন করে ঢোকা মানেই একপিস শেলের আধারে। শেলই সেই আবহ বা অ্যাম্বিয়েন্স, যার বাইরে আপনি বেরোতে পারেন না, বেরোতে পারেন একমাত্র সিস্টেম থেকে বেরিয়ে গেলেই, মানে, লগ-আউট করলেই। কোনো ইউজার লগ-ইন করার মুহূর্তেই চালু হয়ে গেছে প্রথমতম ব্যাশ-প্রক্রিয়াটা। তার পর থেকে যত নতুন নতুন ব্যাশ প্রক্রিয়া চালু করছে ওই ইউজার বা সিস্টেম, তার প্রত্যেকটাই ওই প্রথমতম ব্যাশ প্রক্রিয়ার ছানাপোনা।

এই জনিতৃ প্রক্রিয়া বা পেরেন্ট প্রসেস এবং সন্তান প্রক্রিয়া বা চাইল্ড প্রসেসের ধারণাটা গ্নু-লিনাক্স তথা যে কোনো ইউনিক্স সিস্টেমেই অত্যন্ত গুরুত্বপূর্ণ, এবং অতীব মজার। এবং এর জন্যেও আমার রেফারেন্স ট্যানেনবম। বইটা সত্যিই বেধড়ক। আমার মত যারা অটেকনিকাল রকমে, নিজে নিজে, হাফবয়েল ডিমের মত করে, কম্পিউটার শেখে তাদের জন্যে এর চেয়ে ভালো কোনো বই আমার চোখে পড়েনি। তবে কম্পিউটারের কটা বই-ই বা দেখেছি আমি? লিনাসের সঙ্গে যতই কুচুটেপনা করে থাকুন, মাস্টারমশাই হিশেবে ট্যানেনবম যে পুরো সাবলাইম তাতে কোনো সন্দেহ করে কোন শালা। ট্যানেনবমের স্তরের টেক্সটবই আমি আমার জীবনে আর দুটো মাত্র দেখেছি — স্যামুয়েলসন আর নর্ডহউসের ইকনমিক্স এবং কুরান্টের ক্যালকুলাস। সাসপেন্স থ্রিলারের মত উত্তেজক, টেক্সটবই লিখনা হ্যায় তো অ্যায়সা। একজন পাঠক তথা ছাত্রের গোটা চিন্তাপ্রক্রিয়াটাকেই এতোলবেতোল করে দেওয়া। আমার এই বইটায় বহু জায়গায় বহু বিষয় এসেছে যা সরাসরি ট্যানেনবম থেকে ঝাড়া। মানে যতটা ভালো আমি ঝাড়তে পেরেছি। তবে বইটাকে পড়া দরকার গ্নু-লিনাক্সে নিজের সিস্টেমের সঙ্গে মিলিয়ে মিলিয়ে। ওখানে তো ইউনিক্স উইনডোজ ওএসটু সোলারিস সবই আছে। আর নৈর্ব্যক্তিক একটা রকমে, প্রতিমুহূর্তে সব ধরনের অপারেটিং সিস্টেমের উদাহরণ টেনে টেনে।

যা বলছিলাম, আঞ্চলিক চল বা লোকাল ভ্যারিয়েবল, যাদের সরাসরি, ইন্টারাক্টিভলি, একজন ইউজার বানিয়ে তুলছেন। একটু আগে অ্যালিয়াস দিয়ে যে কাজটা করছিলাম আমরা, এবারে ব্যাশের লোকাল ভ্যারিয়েবল দিয়ে ওই একই কাজ করা যাক। 'likhi' নামের অ্যালিয়াসটা তৈরি করতে গিয়ে আপনি সমান চিহ্নের ডানদিকে যে অংশটা রেখেছিলেন, সেটাকেই এবার একটা ভ্যারিয়েবল বানিয়ে তার মান বা ভ্যালু হিশেবে দিয়ে দিতে হবে। কমান্ড দিন, 'lekho='mpg123 ~/music/*.mp3 & cd ~/probondho;emacs''। দুপাশে দুটো কোট চিহ্ন দিতে হল কারণ মধ্যে স্পেস-চিহ্ন এসেছে। যদি আমরা কোট বা উর্দ্ধ-কমা না-দিতাম, তো, ব্যাশ ভ্যারিয়েবলের মান হিশেবে নিয়ে নিত প্রথম স্পেস চিহ্নের আগে অব্দি। এবার এই যে লোকাল ভ্যারিয়েবল 'lekho' আমরা তৈরি করলাম, এর মান কত বলুন তো? এবং সেই মানটাকে ব্যাশ কী নামে চেনে? আপনার বলতে পারার কথা। না-পারলে চিন্তার কথা, কারণ, তাহলে আরো বহু জরুরি জিনিষই আপনার মনে নেই, আর একবার গোড়া থেকে পড়তে পড়তে আসুন। আমরা এখন 'echo $lekho' কমান্ড দিলেই, ব্যাশ স্ক্রিনে ফুটিয়ে তুলবে সমান চিহ্নের ডানদিকে দুটো কোট চিহ্নের মধ্যে গোটাটা, অবশ্যই কোট ছাড়া। কারণ '$lekho' হল 'lekho' নামের লোকাল ভ্যারিয়েবলের মানের ব্যাশীকৃত নাম। মূল মজাটা অবশ্য অন্যত্র। এবার কমান্ড প্রম্পটে নিছক '$lekho' টাইপ করে এন্টার মারুন। দেখুন, একটু আগে আপনি অ্যালিয়াস দিয়ে যে কাজটা করছিলেন, সেই গোটাটাই এখন এইভাবে করা যাচ্ছে। মানে, আমরা 'lekho' নামের লোকাল ভ্যারিয়েবল দিয়ে সেই কাজটাই করলাম, যেটা একটু আগে 'likhi' নামের অ্যালিয়াস দিয়ে করেছিলাম।

এবার দেখুন, এই সদ্য বানানো লোকাল ভ্যারিয়েবল কিন্তু একদমই এই বিশেষ ব্যাশ প্রক্রিয়ার লোকাল, একে লং-ডিসট্যান্স করবার জন্যে, দূরপাল্লায় ছোটানোর জন্যে, অন্য অন্য ব্যাশ প্রক্রিয়া যারা এর পরে সিস্টেমে শুরু হবে

তাদের কাছে পৌঁছে দেওয়ার জন্যে আপনাকে একে প্রদেশের বাইরে আন্তর্জাতিক স্তরে রপ্তানি করতে হবে। এটা যে একান্তই এই ব্যাশ প্রক্রিয়ার নিজস্ব, সেটা বোঝার উপায় হল, নিছক একবার ‘bash’ টাইপ করে এন্টার মারা। আপনার কমান্ড প্রম্পটে কোনো পার্থক্যই ঘটল না, কিন্তু সিস্টেমের কাছে মোট প্রক্রিয়ার মানচিত্রে আপনার অবস্থানটা বদলে গেল। এতক্ষণ আপনি ব্যাশেই ছিলেন, এখন গেলেন ব্যাশের-মধ্যে-ব্যাশে। এটাও ব্যাশ, কিন্তু একটা নতুন প্রক্রিয়া বা প্রসেস। এখানে আপনি ‘$lekho’ কমান্ড দিন। কী পেলেন? ব্যাশের অজ্ঞতা — কমান্ড নট ফাউন্ড — ব্যাশ এই কমান্ডটা জানেই না। বেচারা জানবেই বা কী করে? আপনি তো রপ্তানি করেননি। এই ব্যাশের-মধ্যে-ব্যাশ থেকে বেরিয়ে যান। তার জন্যে ‘exit’ বা ‘<Ctrl><D>’ তো আছেই। এবার আবার একবার ‘$lekho’ কমান্ড দিয়ে দেখুন, ফের নিখুঁত চিনে যাচ্ছে কমান্ডটা, এমনকি আপনাকে গোটাটা দিতেও হবেনা, ছ-নম্বর দিনের সেই ব্যাশ কমপ্লিশন বা একটু টাইপ করার পর ট্যাব মারলে অন্যটুকু ব্যাশের নিজেই পুরন করে দেওয়ার ব্যাপারটা মনে আছে তো? আপনি কমান্ড প্রম্পটে ‘$le’ টাইপ করে ট্যাব মারুন, হয় দেখবেন গোটা ‘$lekho’-টাই ও দেখিয়ে দিচ্ছে, নয়তো নিচে একাধিক সম্ভাবনা দেখাচ্ছে, যার মধ্যে একটা হল ‘$lekho’। মানে, এই ব্যাশ আমাদের বানানো লোকাল ভ্যারিয়েবলটাকে চেনে। সে তো চিনবেই, এর মধ্যে এমন কিছুই ক্যালি নেই, আমরা তো ভ্যারিয়েবলটা বানিয়েছিলাম এই ব্যাশেই। এবার একে এক্সপোর্ট করুন। কমান্ড দিন ‘export lekho’, এন্টার মারুন। ব্যাস। আপনি রপ্তানি করে দিলেন। এবার আগের মতই ‘bash’ টাইপ করে আর একটা ব্যাশ খুলুন, সেখানে দেখুন, এই ‘$lekho’ কমান্ডটা দিব্য রয়েছে, মানে এক্সপোর্ট হয়ে গেছে এখন থেকে ব্যাশীতব্য যে কোনো ব্যাশের কাছেই।

কিন্তু এই রপ্তানি আপনি সরকারি ভাবে করেননি। রপ্তানি দপ্তরের কোনো বড়কর্তার বা তার জ্যেষ্ঠবাবা কোনো বড়মন্ত্রীর ঘুষ খাওয়ার কোনো সুযোগ আপনি এখনো করেননি, এদিকে রেগুলার দিনে রাতে রপ্তানি করে যাবেন, একি মামদোবাজি নাকি। সরকারি দপ্তরের ফাইল ঘুরিয়ে আনতে হবে এই রপ্তানিকে, যদি যখন তখন লগ-ইন করা মাত্র যে কোনো ব্যাশে আপনি এই ভ্যারিয়েবলটাকে খুঁজে পেতে চান। কিন্তু কোন ফাইল? অবশ্যই ব্যক্তি ইউজার হিশেবে আপনার লোকাল কনফিগারেশন ফাইল। যে লাইনটা দিয়ে আমরা ‘lekho’ ভ্যারিয়েবলটা বানিয়েছিলাম, সেই গোটা লাইনটা, কোট চিহ্ন সহ, নিট যোগ করে দিন আপনার হোমের ‘~/.bashrc’ ফাইলে। এবার ঠিক অ্যালিয়াসের মতই, আপনার লগ-ইন করা থেকে শুরু করে যে কোনো ব্যাশেই পেয়ে যাবেন এই লোকাল ভ্যারিয়েবলটা, এবং তার মান আকারে এই ‘$lekho’ কমান্ডটা। এর আগে অব্দি আমাদের বানানো লোকাল ভ্যারিয়েবল ‘lekho’ ছিল একটা অস্থায়ী ভ্যারিয়েবল এবং একটিমাত্র বিশেষ ব্যাশ-প্রক্রিয়ার নিজস্ব। একটা ব্যাশেই যার জন্ম ও মৃত্যু। এখন এই লোকাল ভ্যারিয়েবলটাকে স্থায়ী করে দেওয়া হল, ‘~/.bashrc’ ফাইলে তার জন্মপ্রক্রিয়াটাকে ভরে দিয়ে। এবার আপনি যদি সিস্টেমের সুপারইউজারও হন, তাহলে এটাকে আপনি সিস্টেম জুড়ে যে কোনো ব্যবহারকারীর কাছেই প্রাপ্তব্য করে তুলতে পারেন। তখন আপনি আর একজন ইউজার হয়ে নিজের হোম ডিরেক্টরির ‘~/.bashrc’ ফাইলে লাইনটা যোগ করবেন না। যোগ করবেন ব্যাশের গ্লোবাল কনফিগারেশন ফাইলে। সেরকম কোনো ফাইলের কথা আপনি জানেন নাকি? গ্লোবাল কনফিগারেশন ফাইলে যোগ করে দিলে তখন সেটা হয়ে যাবে গ্লোবাল ভ্যারিয়েবল, সিস্টেম জুড়ে যে কোনো ইউজারের বেলাতেই সত্যি। এবং স্থায়ী তো বটেই। যে কোনো ইউজারের যে কোনো ব্যাশ চালু হওয়ার আগে ‘/etc’ ডিরেক্টরির কনফিগারেশন ফাইল পড়েই তাদের বানিয়ে নেবে। তখন সেটা মোট এনভায়রনমেন্টের বা প্রতিবেশেরই একটা অংশ হয়ে যাবে, তাকে আমরা ডাকব প্রতিবেশ চল বা এনভায়রনমেন্ট ভ্যারিয়েবল বলে।

ব্যাশের কাজ করার এই দুই ধরনের ভ্যারিয়েবল — এনভায়রনমেন্ট ভ্যারিয়েবল এবং লোকাল ভ্যারিয়েবল এরা ব্যাশের কাছে আসে দুই ভাবে। এনভায়রনমেন্ট ভ্যারিয়েবল বা প্রতিবেশ-চলদের গজিয়ে তোলে সিস্টেম, ‘/etc/profiles’ বা ‘/etc/bashrc’ গোছের গ্লোবাল কনফিগারেশন ফাইল অনুযায়ী। এদের মধ্যে পড়ে ‘SHELL’, ‘PS1’, ‘PATH’ ইত্যাদি। পরে আমরা বিশদ ভাবে আসছি এদের কথায়। আর লোকাল ভ্যারিয়েবল বা স্থানীয়-চলদের বানিয়ে তোলে ইউজার, একটা জ্যান্ত পারস্পরিক ইন্টারাক্টিভ ব্যাশ প্রক্রিয়ায়, নিজে কমান্ড প্রম্পটে কমান্ড দিয়ে, যেমন আমরা ‘lekho’-কে বানালাম। বা একজন ইউজারের নিজের হোম ডিরেক্টরিতে থাকা লোকাল কনফিগারেশন ফাইল পড়ে সেই ইউজারের জন্যে চালু হওয়া ব্যাশ-প্রক্রিয়া এদের বানিয়ে তোলে। এই লোকাল ভ্যারিয়েবলদের হালহদিশ ভরা থাকে ইউজারের নিজের ‘~/.profile’ বা ‘~/.bashrc’ গোছের ফাইলে। ব্যাশ চালু হওয়ার সময় এই

ব্যক্তি ইউজারের লোকাল কনফিগারেশন অনুযায়ী এদের বানায়, যে কনফিগারেশনগুলো নিজের মর্জি বা প্রয়োজন মত বদলে নেয় একজন ব্যক্তি ব্যবহারকারী। ঠিক যে ভাবে পরে আমরা 'lekho' বানানোর আদেশকে স্থায়ী করার জন্যে কনফিগারেশন ফাইলে ভরে দিয়েছিলাম।

৬।। ব্যাশ ভ্যারিয়েবলের ঠিকুজি

একটু আগে কমান্ড প্রম্পটে যে ধরনের কমান্ড দিয়ে আমরা 'lekho' ভ্যারিয়েবলটা বানালাম, একে বলে ভ্যারিয়েবলের ডেফিনিশন বা সংজ্ঞা। এই চল-সংজ্ঞা বা ভ্যারিয়েবল-ডেফিনিশনের তিনটে অংশ থাকে। প্রথম অংশটা তার নাম, ধরুন, '*variable_name*'। তার পরে একটা অ্যাসাইনমেন্ট অপারেটর বা ধার্যীকরণ চিহ্ন মানে আমাদের চেনা সমান চিহ্ন '='। তারপর ভ্যারিয়েবলের মান, ধরুন, '*variable_value*'। এখানে আমরা এই '=' চিহ্নকে অ্যাসাইনমেন্ট অপারেটর বলে ডাকছি কেন, সেটা খেয়াল করুন, পরে, শুধু শেল-স্ক্রিপ্টে নয়, যে কোনো কম্পিউটার ভাষাকে বোঝার জন্যেই আপনার এটা কাজে লাগবে। যখনই ভ্যারিয়েবলের নামটা দিলাম, সেই নামে স্মৃতিভূমিকে একখণ্ড জমি চিহ্নিত হল। এবার অ্যাসাইনমেন্ট অপারেটর ব্যাশকে জানাল যে এবার ডানদিকে যে ভ্যারিয়েবলের মান দেওয়া হচ্ছে সেটাকে অ্যাসাইন বা ধার্য করে দিতে হবে, ভরে দিতে হবে ওই জমিটুকুতে। শুধু মাথায় রাখবেন, '=' চিহ্নের আগে এবং পরে কোনো স্পেস দেবেন না। স্পেস দিলে ব্যাশ আর একে অ্যাসাইনমেন্ট অপারেটর বলে চিনতে পারবে না। সংজ্ঞার কাঠামোটা হল — '*variable_name=variable_value*'। এই কাঠামোটার সঙ্গে মিলিয়ে নিন, একটু আগে আমরা যে 'lekho' ভ্যারিয়েবলটা বানিয়েছিলাম, তার সংজ্ঞাটাকে। '*variable_name*' হল জমি, আর '*variable_value*' তার ফলন, এই প্রতিতুলনাটা আমরা আগেই দিয়েছিলাম।

আর এই ব্যাশের বাইরে অন্য যে কোনো ব্যাশ প্রক্রিয়াতেও বা অন্য যে কোনো প্রোগ্রামের কাছেও এই চলটা হাজির করার কৌশলটা তো একটু আগেই দেখলাম আমরা, এক্সপোর্ট করার। '`export` *variable_name*' হল এই রপ্তানি আদেশের কাঠামো। মনে করে দেখুন, কী ভাবে আমরা 'lekho' চলটাকে রপ্তানি করেছিলাম যে কোনো ব্যাশ তথা যে কোনো প্রোগ্রামের কাছে। এই ব্যাশ বা এই প্রোগ্রাম সবগুলোই কিন্তু এই ইউজারের, কারণ এটা আঞ্চলিক চল বা লোকাল ভ্যারিয়েবল। গ্লোবালি মানে গোটা সিস্টেম জুড়ে একটা ভ্যারিয়েবলকে কী করে রুট হাজির করতে পারে, সেটাও জেনেছি আমরা।

এবার, যেই ভ্যারিয়েবলের সংজ্ঞা দিয়ে সেটাকে বানানো এবং তারপর সেটাকে যে কোনো প্রোগ্রামের কাছে রপ্তানি করা হয়ে গেল, তখন এই ইউজারের চালানো যে কোনো প্রোগ্রাম চলটাকে ব্যবহার করতে পারবে। ব্যবহার করতে গেলে তাকে এই ভ্যারিয়েবলের নামটা দিয়ে উল্লেখ করতে হবে, এবং নামের আগে লাগাতে হবে ওই '$' চিহ্নটা, যা দিয়ে আমরা যে কোনো ভ্যারিয়েবলের মানকে বোঝাই। যেমন করে একটু আগে আমরা ব্যবহার করলাম 'lekho' চলটাকে। যে কোনো মুহূর্তে কোনো ভ্যারিয়েবলের জমিতে কোন মান ফলে আছে তা জানার উপায় হল '`echo`'। আদেশ দিন, '`echo $`*variable_name*'। দেখুন স্ক্রিনে তার মানটা, '*variable_value*', ফুটে উঠেছে। যেমন, '`echo $lekho`' আদেশ দিলে আমরা পাব ওই গোটা লাইনটা। দিয়ে দেখুন।

এই ভ্যারিয়েবলগুলোর কনফিগারেশন কী ভাবে আমরা বদলাব তার প্রথমতম কায়দাটা আমরা তো জানিই, কনফিগারেশন ফাইলে কোনো একটা লাইনকে যদি আমরা অকেজো করে দিতে চাই, তাহলে, তার আগে একটা '#' চিহ্ন লাগিয়ে দেওয়া। এটা নানা জায়গাতেই বেশ কাজে আসে। একটা লাইন কী কাজ করছে সেটা বোঝার জন্যে লাইনটার আগে একটা '#' চিহ্ন লাগিয়ে দিলাম। পরে, যদি ফেরত আসার দরকার পড়ে, '#' চিহ্নটাকে উড়িয়ে দিলাম। একে বলে কমেন্টিং আনকমেন্টিং। মন্তব্যীকরণ অমন্তব্যীকরণ। 'মন্তব্য' মানে যা মূল ফাইল নয়, ফাইলেরই কোনো বিষয় নিয়ে কোনো আলোচনা। '#' চিহ্নটা থাকলে ব্যাশ ধরে নেবে, লাইনটা একটা মন্তব্য, আর পড়বেনা। এই মন্তব্যীকরণ আর একটা কারণেও দরকার পড়ে, সেটা হল, একটা কনফিগারেশন ফাইলে বা একটা শেল স্ক্রিপ্টে কোথায় কী করা হচ্ছে সেটার একটা হদিশ দিয়ে দেওয়া। যেমন, '/etc' ডিরেক্টরিতে অনেকগুলো কনফিগারেশন ফাইলেরই কোন অংশটা আসলে কী কাজ করে, সেটা আমরা জানতে পেরেছিলাম ফাইলগুলোর '#' চিহ্ন লাগানো লাইন মানে কমেন্ট লাইনগুলো পড়ে। শুধু এই জন্যে না, যে স্ক্রিপ্টগুলো আপনি নিজে বানাবেন, সেগুলোতেও এটা লাগানো ভারি পুণ্যের কাজ। শুধু অন্যের জন্যে না, নিজের জন্যেও। এটা অবশ্যম্ভাবী যে প্রথম যেদিন আপনি শেল স্ক্রিপ্ট বানাতে শুরু করবেন, প্রথম স্ক্রিপ্ট শেখার ধুমকিতে আপনি যেখান থেকে শিখছেন সেখানে দেওয়া প্রতিটি

নিয়ম নানা ভাবে প্রয়োগ করে নানা স্ক্রিপ্ট লিখবেন, যার বেশির ভাগ কলকব্জাই আপনি মেশিন অফ করে বেরোনোর পরে বেমালুম ভুলে যাবেন, আর পরে কখনো, স্ক্রিপ্টটা দেখে, আপনাকে মাথার চুল ছিঁড়তে হবে এটা বুঝতে যে স্ক্রিপ্টটা কী করে, বেড়াল তাড়ায় না ঝাড়ে বাঁশ কাটে। তাই, নিজেকে এবং অন্যকে এত কষ্ট দেওয়ার চেয়ে প্রথম থেকেই কমেন্ট লাগানো প্র্যাকটিস করুন। স্টিভ উয়ালিনের সি প্রোগ্রামিং নিয়ে বইটায় সি প্রোগ্রাম লেখার এবং কমেন্ট লেখার ধাঁচ নিয়ে ভারি মজার কিছু আলোচনা আছে, পড়ে দেখতে পারেন, সেটা শেল স্ক্রিপ্ট নিয়েও একই রকম খাটে। সিস্টেমের মধ্যে ব্যাশ ব্যাশ স্ক্রিপ্টিং নিয়ে অর্বুদ অর্বুদ মালপত্তর আছে, নেট থেকেও পাবেন, জিএলটির সিডিতেও আমরা দিই। দরকার হলে যোগাযোগ করবেন। জিএলটির মেল আইডি তো দিয়েছি, আমারটাও দিয়ে দিই, ‘paagol@softhome.net’। এই দেড় লাখ শব্দ জুড়ে আমার লেখা পড়তে পড়তে আপনি তো আমাকেও কিছুটা মেপে নিয়েছেন, সেটা বেশ মিলে যাচ্ছেনা? দেখেছেন, স্বতঃব্যাখ্যাশীল ইমেল আইডি মনে রাখার কী সুবিধে। ঠিক একই জিনিষ মনে রাখবেন ভ্যারিয়েবলগুলো নিয়েও, কম টাইপ করার স্বার্থে আপনি খুব ছোট ছোট ভ্যারিয়েবল নাম লাগাতেই পারেন, ঠেলাটা বুঝবেন পরে, যখন কিছুতেই মনে করতে পারবেন না, কোন ভ্যারিয়েবলটা কী। কিন্তু নামটা থেকেই যদি আন্দাজ করা যায়, এই ঝামেলাটাই হয়না। এটাও, শুধু শেল স্ক্রিপ্ট না, যে কোনো প্রোগ্রামিং-এর জন্যেই সত্যি। আর, আরো একবার বলছি, মোটামুটি পরিমাণ কম্পিউটার ব্যবহার করার প্ল্যান থাকলেও, টাচ টাইপিং শিখে নিন, মানে দশ আঙুলে টাইপ করা। প্রথম বছরেই আপনার এতটা সময় বাঁচবে মোট কাজের যে প্রাথমিক পরিশ্রমটা উশুল হয়ে যাবে। গ্নু-লিনাক্সে টাইপ শেখার প্যাকেজ জিটাইপিস্ট (gtypist) — আগেই বলেছি, ভীষণ কাজের।

৭।। বিভিন্ন ব্যাশ ভ্যারিয়েবলের কনফিগারেশন

ব্যাশ ভ্যারিয়েবল কনফিগার করতে নেমে আমাদের প্রথম মুরগী সেই পুরোনো পাপী পথনির্দেশ — ‘PATH’। বারবার নানা কাজে নানা প্রসঙ্গে আমরা তাকে টেনে এনেছি, বদলে যেতে দেখেছি, কিন্তু নিজেরা হাতে করে কখনো বদলাইনি। কিন্তু বদলানোর প্রয়োজনটা প্রায়ই হয়, কখনো কখনো সাময়িক, কখনো স্থায়ী রকমে। ধরুন, কোনো একটা ফাইল বা প্রোগ্রাম আপনি আপনার কাজে লাগাতে চাইছেন, যা স্বাভাবিক ভাবে আপনার পথনির্দেশে নেই। একটা উদাহরণ নেওয়া যাক। প্রথমে আমার মেশিনের সুজে সিস্টেমে ‘dd’ নামের ইউজার হিশেবে আমার পথনির্দেশটা জেনে নেওয়া যাক। এর কমান্ডটা আমরা জানি, ‘`echo $PATH`’। স্ক্রিনে ফুটে ওঠা এর ফলাফলটা তুলে দেওয়া যাক।

```
/home/dd/bin: /usr/local/bin: /usr/bin: /usr/X11R6/bin: /bin: /usr/games:
/opt/gnome2/bin: /opt/gnome/bin: /opt/kde3/bin: /usr/lib/java/jre/bin:
/opt/gnome/bin
```

এর মধ্যে দেখুন আলাদা আলাদা এগারোটা উপাদান আছে। এগারোটা আলাদা আলাদা পথ। তাদের সমাহারটা হল ‘dd’ নামক ইউজারের জন্যে গোটা পথনির্দেশ। এই উপাদানগুলো পথনির্দেশের মধ্যে থাকে পরপর, শুধু মধ্যে একটা করে যতিচিহ্ন ‘:’। কোনো স্পেস থাকেনা। প্রতিটি কোলনের পরে একটা করে স্পেস আমি এখন আনলাম, নইলে যেখানে সেখানে লাইন ভেঙে দিচ্ছে, আর বুঝতেও অসুবিধে হতে পারে একটু অনভ্যস্ত চোখে। প্রথম উপাদানটা দেখুন, ‘dd’ নামক ইউজারের নিজের হোম হল ‘/home/dd’। তার মধ্যে একটা বাইনারি ডিরেক্টরি, ‘/home/dd/bin’। এই ডিরেক্টরিটা কিন্তু এমনিতে থাকেনা হোম ডিরেক্টরিতে। মানে সুজে বা স্ল্যাকওয়ার কোনোটাতেই থাকেনা। কিন্তু পথনির্দেশে থাকা মানে এই ইউজার চাইলেই বানিয়ে নিতে পারে ডিরেক্টরিটা। তার নিজের বানানো কোনো প্রোগ্রাম বা শেল-স্ক্রিপ্ট সে যদি এখানে রাখে সেটা শেল পেয়ে যাবে। আলাদা করে প্রোগ্রামটা কোথায় আছে তার পুরো পথটা দিয়ে দিতে হবেনা। যেমন নয় নম্বর দিনে বারবার ‘writefiles’ ইত্যাদি যে শেল-স্ক্রিপ্টগুলো আমরা ব্যবহার করেছি, তাদের সবগুলোই রাখা আছে ‘dd’-র নিজের বানিয়ে নেওয়া এই ‘/home/dd/bin’ ডিরেক্টরিতে। তাই কমান্ড প্রম্পটে জাস্ট ‘`writefiles`’ টাইপ করে দিলেই স্ক্রিপ্টটা চালাতে পারে ব্যাশ। কিন্তু যদি না-থাকত তখন কী হত?

গুইতে বা গ্রাফিকাল ইউজার ইন্টারফেসে আমি যখন জজজ-এ, মানে ইন্টারনেটে ব্রাউজ বা ঘোরাঘুরি করি, যে ব্রাউজারটা সবচেয়ে বেশি ব্যবহার করি সেটা হল মোজিলা-এক্সএফটি (mozilla-xft)। এই প্যাকেজটা এমনিতে থাকেনা সিস্টেমে, অরিজিত বলেছিল, এটা সুজের জন্যে খুব ভালো কাজ করবে। নেট থেকে নামিয়েছিলাম প্যাকেজটা। টার করা, বিজিপটু কোঁকড়ানো থেকে বড় করে নিয়েছিলাম ‘/opt’ ডিরেক্টরির মধ্যে ‘personal’ বলে

একটা সাবডিরেক্টরিতে, ‘tar xvjf’ কমান্ড দিয়ে। এতে ‘/opt/personal/mozilla’ নামে একটা ডিরেক্টরি তৈরি হয়ে গেছিল। ‘/opt/personal’ ডিরেক্টরিতে না-করে সরাসরি ‘/opt’ ডিরেক্টরিতেও করা যেত, কিন্তু সেখানে স্বাভাবিক মোজিলার জন্যে আগে থেকেই একটা ‘/opt/mozilla’ বলে ডিরেক্টরি আছে। মোজিলা-এক্সএফটি আমায় কম্পাইল করতে হয়নি, এটা একটা প্রিকম্পাইলড বাইনারি। মানে এর ভিতরে ‘mozilla-xft’ নামে একটা বাইনারি বা প্রোগ্রাম ফাইল আগে থেকেই তৈরি হয়ে আছে, যেটা সুজে ৮.২ সিস্টেমের মত করেই তৈরি। এবার সমস্যাটা হল এই যে এই বাইনারিটার ঠিকানাটা দেখুন — ‘/opt/personal/mozilla/mozill-xft’ — এই পথটা কিন্তু আমার ‘$PATH’-এ নেই। তাহলে আমি কী করব?

একটা আছে ব্রুট ফোর্স বা গাজোয়ারি পন্থা। চালাচ্ছি যখন সরাসরি গোটা পথটা দিয়ে দেওয়া। মানে যখনই মোজিলা-এক্সএফটি চালাচ্ছি, তখনই তার গোটা পথটা দিয়ে দেওয়া — ‘/opt/personal/mozilla/mozilla-xft’। নইলে ব্যাশ প্রতিবারই একবার করে তার অজ্ঞতা জ্ঞাপন করবে, কমান্ড নট ফাউন্ড। প্রতিবারই এরকম দিতে হবে যতবার আমরা মোজিলা-এক্সএফটি চালাব। এর একটা সহজ উপায় হল, সাময়িক ভাবে ‘PATH’ ভ্যারিয়েবলটাকে লোকালি কনফিগার করে নেওয়া। এর জন্যে কমান্ড লাগবে দুটো। পরপর।

```
PATH=$PATH:/opt/personal/mozilla
export PATH
```

কমান্ড দুটোর দ্বিতীয়টাকে আমরা ইতিমধ্যেই চিনি। রপ্তানি করে অন্য ব্যাশ তথা প্রোগ্রামের কাছে হাজির করে দেওয়া। প্রথম কমান্ডটাই আসলে চালু ‘PATH’ ভ্যারিয়েবলটাকে আমাদের প্রয়োজন মোতাবেক বদলে দিচ্ছে। আমরা জানি ‘=’ একটা অ্যাসাইনমেন্ট অপারেটর, একটা বিশেষ মানকে অ্যাসাইন করে দেয়, লাগু করে দেয় একটা বিশেষ ভ্যারিয়েবলের জমিতে। এই ‘=’ চিহ্নের বাঁদিকে থাকে ভ্যারিয়েবলের নাম, ডানদিকে থাকে আমাদের কাঙ্খিত মান। একটু আগে ভ্যারিয়েবলের সংজ্ঞা বা ডেফিনিশনের স্টেটমেন্ট থেকে দেখে নিন। তার মানে, এই চিহ্নের বাঁদিকে আছে ‘PATH’ নামের ভ্যারিয়েবল, আর ডানদিকে তার যে ভ্যালুটা আমরা চাইছি, মানে নতুন পরিবর্তিত ‘$PATH’। এবার ডানদিকের পুরোটার মধ্যে দেখুন, দুটো অংশ। তাদের মধ্যে আমাদের এইমাত্র পরিচিত যতিচিহ্ন ‘:’, যে চিহ্নটা, এইমাত্র আমরা দেখেছি, পরপর একাধিক অংশকে একটার সঙ্গে আর একটা যোগ করে যায় ‘$PATH’ মানে ‘PATH’ ভ্যারিয়েবলের মান বোঝাতে গিয়ে। এখানেও ‘:’ চিহ্নটা ঠিক তাই করছে। নতুন ‘$PATH’-এ দুটো অংশকে জুড়ে দিচ্ছে। পুরোনো ‘$PATH’, মানে কমান্ড শুরু হওয়ার সময়ে ‘$PATH’ বলতে ব্যাশ যা বোঝে, যাকে সে এখানে উল্লেখ করছে ‘$PATH’ বলে, এবং তার পরে ওই ‘:’ চিহ্ন, সবশেষে সেই অতিরিক্ত পথাংশ যা আমরা চাইছি, মানে ‘/opt/personal/mozilla’। তার মানে দেখুন, এই কমান্ডটা তার যাত্রা শুরু করছে একটা ‘$PATH’ নিয়ে, এবং শেষ হচ্ছে আর একটা ‘$PATH’ নিয়ে। এই নতুন ‘$PATH’ হল পুরোনো ‘$PATH’ যোগ ওই বাড়তি পথাংশ ‘/opt/personal/mozilla’।

এবার, নতুন ‘$PATH’ যে সত্যিই বদলে গেছে সেটা আমরা প্রমাণ পাব কী করে? খুব সোজা, আবার সেই ব্যাশকে প্রতিধবনি করতে বলা, ‘echo $PATH’। এবার যে মানটা পেলাম আমরা সেটায় ওই পথাংশ যোগ হয়ে গেছে।

```
/home/dd/bin: /usr/local/bin: /usr/bin: /usr/X11R6/bin: /bin: /usr/games:
/opt/gnome2/bin: /opt/gnome/bin: /opt/kde3/bin: /usr/lib/java/jre/bin:
/opt/gnome/bin: /opt/personal/mozilla
```

এখন দেখুন, মোট পথাংশ এগারোটার জায়গায় বারোটা হয়ে গেছে। তার বারোতমটা হল আমাদের যোগ করা। ঠিক যেমন আমরা দেখেছিলাম, আগে রয়েছে পুরোনো ‘$PATH’, তার পর এসে গেছে আমাদের যোগ করা অংশটা। এতে লাভটা কী হয়েছে? এখন আমি যেই কমান্ড প্রম্পটে ‘mozilla-xft’ কমান্ডটা ব্যবহার করব, ব্যাশ আর বলবে না, আমি কী জানি। সে পরপর খুঁজতে খুঁজতে বারো নম্বর অংশটা খোঁজার সময়েই পেয়ে যাবে তার আরব্ধ কমান্ড। মানে আমাদের চাওয়া ওই পথাংশ আমার ব্যাশের ‘$PATH’-এ এসে গেছে। এই বদলটা কিন্তু সাময়িক। একবার লগ-আউট করে গেলে পরের বার এটা আর পাওয়া যাবেনা। তখন ব্যাশ আবার ‘$PATH’ বলতে প্রথম মানটাই বুঝবে। এবার, আপনি অফিসিয়াল রকমে এই বদলটা যদি পাকাপোক্ত করে নিতে চান, তার উপায় কী?

এরকম কোনো উপায়ের কথা আপনি কি জানেন? যদি না-জানেন, করবেন না, সিম্পল। আমি কী জানি? এটা শুধু ব্যাশবাবু নয়, আমাদের লাগের আঁফা তারিবল সায়মিন্দুর ক্যাচফ্রেজ।

চিকেন-ফ্লু উবে গেছে, এবার একটা মুরগী মানে পথনির্দেশ জবাই করেই থামলে চলবে? মুরগী থেকে মুরগী, সভ্যতা তো এভাবেই এগিয়েছে। সভ্যতার ইতিহাসের দিকে তাকিয়ে দেখুন, ক্রমউন্নতিশীল মুরগী জবাইয়ের কালপথরেখা। আমাদের দ্বিতীয় মুরগী নেওয়া যাক প্রম্পট, কমান্ড প্রম্পট। প্রতিমুহূর্তে যার মুখ আমাদের দেখতে হচ্ছে। মুখ মানে কালো মুখ, তাতে সাদা আখরে ফুটিয়ে তোলা প্রম্পট। যারা পড়ছেন, তার মধ্যে আমার মত বুড়ো যদি কেউ থাকেন, তারা তাদের একসময়কার পুরোনো ডস আর উইনডোজের অভিজ্ঞতায় ‘autoexec.bat’ নামে ব্যাচ ফাইলের সঙ্গে মিল পাচ্ছেন নিশ্চয়ই। ব্যাচ ফাইলকে একধরনের দুধেভাতে শেল স্ক্রিপ্ট বলতে পারেন। সেখানেও মনে করে দেখুন, প্রথম দুটো বিষয়ই থাকত এই পথ আর প্রম্পট। তার পরেই আসত কটা ফাইল একসঙ্গে খোলা যাবে তার হদিশ। সেসব বিটলেপনা গ্নু-লিনাক্সে নেই। এখানে মেমরি এবং ভারচুয়াল মেমরি ওরকম খাজা রকমে নিয়ন্ত্রণ করা হয়না। প্রথম প্রথম উইনডোজ থেকে গ্নু-লিনাক্সে এসে চমকটা লাগে এখানেই — একসঙ্গে এতকিছু এতগুলো কাজ এত চমৎকার করা যায়, এই মেশিনেই, কই উইনডোজে তো হতনা? এর কারণ, মাল্টিটাস্কিং-টা ডস তথা উইনডোজ সিস্টেমে আরোপিত, প্রক্ষিপ্ত, বাইরে থেকে চাপানো। বহরমপুরের সেই ভদ্রলোকের মত, যিনি মাঝেমাঝেই কয়েকটা করে রবীন্দ্রসঙ্গীত লিখে রেডিওস্টেশনে পাঠিয়ে দিতেন, এবং এদের একচোখোমি দেখে চটে যেতেন, দেখেছ, তিনি মফস্বলে থাকেন বলে তার রবীন্দ্রসঙ্গীত এরা বাজাচ্ছে না। আমাদের চার আর পাঁচ নম্বর দিনের ইউনিক্স বিবর্তনের ইতিহাসের সঙ্গে দু নম্বর দিনের মাল্টিপ্লেক্সিং-এর আলোচনাটাকে আর আট নম্বর দিনের ভার্চুয়াল মেমরি ফাইল সিস্টেমের আলোচনাটাকে মিলিয়ে ভাবুন, তাহলেই বুঝতে পারবেন, ইউনিক্স মাল্টিপ্লেক্সিং-এর মধ্যে দিয়েই তৈরি হয়েছে। এক সঙ্গে অনেক ব্যবহারকারীর অনেক রকম প্রক্রিয়াকে মেলাবেন তিনি মেলাবেন বলেই তৈরি হয়েছিল ইউনিক্স, এবং তারই উত্তরাধিকারে ওপনসোর্স ইউনিক্স মানে গ্নু-লিনাক্স। তাই ঝোড়ো হাওয়া আর পোড়ো বাড়িটার ওই ভাঙা দরজাটা সেখানে জোর করে মেলাতে হয়না, যাতে একটা গান বাজাতে শুরু করলেও ডিফ্র্যাগমেন্টেশন শুধুই বারবার কোঁত পাড়তে থাকে, নতুন করে শুরু হয়, নিজেই গাইতে শুরু করে দেয়, আমারে তুমি অশেষ করেছো। অন্য কিছুর সঙ্গে একই সাথে সিনেমা টিনেমার কথা তো ছেড়েই দিন। যেখানে একটা আপডেট-ডিবি এবং একটা ব্যাকআপ পিছনে ব্যাকগ্রাউন্ড প্রসেসে নিয়ে গ্নু-লিনাক্সে দিব্য আমি, এই গতকালই, একই সঙ্গে পাশাপাশি ম্যাট্রিক্স সিনেমাটা চালিয়ে, পাশে তার চিত্রনাট্যটা খোলা, আর পাশে একটা ইম্যাক্সে তার নোট নিচ্ছিলাম। কিয়ানু রিভস এমন একটা হেঁচকি অব্দি তুলল না যা চিত্রনাট্যে নেই, মাইরি, মিলিয়ে বলছি। চিত্রনাট্যটা নেট থেকে নামানো, অরিজিনাল।

হ্যাঁ, প্রম্পট। এই প্রম্পট করার কেউ নেই বলেই, মূল পার্ট ভুলে গিয়ে মাঝে মাঝেই বাজে বকতে থাকছি। যাকগে এখন আর ভেবে লাভ নেই, বইই শেষ হয়ে এল। এক লাখ পঁয়তাল্লিশ হাজার শব্দ নামিয়ে দিয়েছি, আর হাজার পাঁচেকের মধ্যেই ছুটি, সেই নভেম্বরের শুরু থেকে আজ মার্চের তিন, আমাদের লেঠেল বংশের জিনে দম প্রচুর, তাও, সেটাও ফুরিয়ে আসছে এবার। আর আপনার কথা ভেবে, সত্যি বলছি, আমার নিজেরি কান্না পাচ্ছে। যাইহোক, আপনার প্রম্পটের একটা আকার আছে। যেখানে আপনি কমান্ড দিচ্ছেন। আমরা যেমন পাঁচ নম্বর দিনে ম্যানড্রেকের কমান্ড প্রম্পটটা দেখিয়েছিলাম। ডিস্ট্রো থেকে ডিস্ট্রোয় সেটা একটু আধটু তফাত হয়। ডিফল্ট সেটিং-এ। সেটাকে নিজের মত করে বদলেও নেওয়া যায়। এই কাজে যে ভ্যারিয়েবলটা ব্যবহার করা হয় তার নাম ‘PS1’। ‘PATH’-এর মত আর একটা এনভায়রনমেন্ট ভ্যারিয়েবল। তবে একেও আপনি লোকালি নিজের মত করে বদলে নিতে পারেন। সিস্টেমে অন্য ইউজারের প্রম্পট তাতে বদলাবে না, শুধু আপনারটা বদলে যাবে। যেমন, সুজে সিস্টেমে প্রম্পটটা দেখুন, এটা ইউজার ‘dd’-র জন্যে। অন্য ইউজারে ওই অংশটা বদলে যাবে।

```
dd@mahammad:~/Documents/lesson>
```

এবার দেখা যাক ‘PS1’ ভ্যারিয়েবলের কী মান কমান্ড প্রম্পটটাকে এই আকার দিচ্ছে। তার জন্যে আমরা কমান্ডটা জানি ‘`echo $PS1`’। এই কমান্ড দেওয়ায় ব্যাশ স্ক্রিনে ফুটিয়ে তুলল নিচের এই শব্দবন্ধ। হঠাৎ করে দেখলে মনে হতেই পারে, ব্যাশ আমায় হিব্রু ভাষায় গালাগাল দিচ্ছে। এত খাটাই যে।

```
\u@\h:\w>
```

এই ‘$PS1’-এর শেষ ‘>’ চিহ্নটা কমান্ড প্রম্পটের সঙ্গে মিলিয়ে নিন। দুটোতেই এক। এবার দেখুন, আরো একটা চিহ্ন একদম এক। সেটা হল ‘@’। মানে অ্যাকাউন্টান্সিতে যা ছিল ‘অ্যাট-দি-রেট-অফ’, এখন নেট পরিভাষায় ‘অ্যাট’। এখানে ঠিক ‘অ্যাট’ এই অর্থেই ব্যবহার হচ্ছে। এবং নেটের লিংকগুলো, ওয়েবসাইটের মধ্যেকার বিভিন্ন ঠিকানা এমনকি উইনডোজেও দেখায়, খেয়াল করবেন, ‘/’ যতিচিহ্ন ব্যবহার করে, উইনডোজের পথনির্দেশের স্বাভাবিক যতিচিহ্ন কিন্তু ‘\’, ছয় নম্বর দিনের সেকশন ৪ থেকে দেখুন। আসলে এই নেট ওয়েব তার বিভিন্ন ক্রিয়াকলাপের ব্যাকরণ — এই গোটাটাই গজিয়ে উঠেছে ইউনিক্স জগত থেকে। সেই ব্যাকরণ মেনেই, নিজের সিস্টেমে যে পথনির্দেশই দেখা, ওয়েবসাইটের বেলায় উইনডোজকেও সেটা মানতে হয়। সেটা বোঝা যাবে এর ঠিক আগের আর পরের ‘\u’ আর ‘\h’ অংশদুটোর মানে জানলে। ম্যানুয়াল দেখে তো এখনি জেনে যেতে পারেন, সেটা হল জানার বোঝার সাহেব রকম, সব কিছু জেনেই যাদের শিখতে এবং বুঝতে হয়। আমরা তো বহুসময় বইপত্তরই পাই না, তাই, বাধ্যতই, নিজেকেই চেষ্টা করতে হয়। আমরা জানি বাঁদিকের অংশটা যে ইউজারের ব্যাশ তার নাম, সেখান থেকে আন্দাজ করা যায় ‘\u’ মানে নিশ্চয়ই ইউজারের নাম। এবার বাকি রইল ‘\h’। প্রম্পটে যার আকার হল ‘mahammad’। নয় নম্বর দিনে, মনে করে দেখুন তো, এই নামটা পেয়েছিলেন? পেয়েছিলেন হোস্টনেম কনফিগারেশনের ফাইলের সূত্রে। এবার হোস্টনেমের এইচ আর এই এইচ, আর যায় কোথায়? নিশ্চয়ই এটা হোস্টনেম। আর তারপরে একটা ‘~’ চিহ্ন। এটা নিশ্চিতভাবেই হোম-ডিরেক্টরি, ব্যাশ ‘~’ বলতেই চালু ইউজারের হোম ডিরেক্টরি বোঝে। এবার মিলিয়ে দেখুন। আবার সেই ব্যাশের ম্যানুয়াল। একটা আস্ত সেকশন আছে দেখুন, ‘PROMPTING’। সেই সেকশন থেকে এই রকম কয়েকটা চিহ্নের মানে তুলে দেওয়া যাক।

```
\d     the date in "Weekday Month Date" format (e.g., "Tue May 26")
\h     the hostname up to the first `.'
\t     the current time in 24-hour HH:MM:SS format
\u     the username of the current user
\w     the current working directory
```

মিলিয়ে দেখুন, একটু আমাদের প্রম্পটের আকারের সঙ্গে ‘PS1’ ভ্যারিয়েবলের যে আকারটা আমরা পেলাম, ‘\u@\h:\w>’, সেটাকে বুঝতে পারছেন কিনা। এবার, যেই আমরা ফরমুলাগুলো দেখে ফেললাম, আমাদের প্রম্পট বদলানো আটকায় কার পিতাশ্রী? কমান্ড দেওয়া যাক, ‘PS1="\u@\h<\w>\t>"’, যাতে ফরমুলা মোতাবেক আগের প্রম্পটের উপাদানগুলোর এখন সময়টাও দেখায়। ঠিক তাই হল। প্রম্পটের আকার হয়ে গেল —

```
dd@mahammad<~>23:07:15>
```

এবার চালু কাজের ডিরেক্টরিটাকে আমরা দুপাশে দুটো ‘<’ আর ‘>’ চিহ্নের মধ্যে দিতে বলেছিলাম, ঠিক তাই করেছে ব্যাশ। আর তারপর দিতে বলেছিলাম ‘\t’, মানে সময়, চব্বিশ ঘন্টার ফরম্যাটে, মানে দুপুর একটা যেখানে তেরোটা, ট্রেনের সময়ের মত। এরকম আর যা যা আপনার মন চায় দিতে পারেন। অজস্র অপশন আছে ব্যাশের ম্যানুয়ালে। আমি ঠিক দরকার যে কটা তার বাইরে আর একটাও দিইনি। মানে কয়েকটা দিয়ে ফেলেছিলাম, খেয়াল করে আবার উড়িয়ে দিলাম। এখান থেকেই পড়ে করে নেবেন, সে চলবেনা। নিজে বার করে নিজে পড়ুন। প্রম্পটের এই অস্থায়ী বদলকে স্থায়ী করবেন কী করে? ঠিক ‘PATH’ ভ্যারিয়েবলের মত, ‘PS1’ ভ্যারিয়েবলও এই ইউজারের ব্যাশে স্থায়ী হবে তখনই যখন একে লোকাল কনফিগারেশনে তুলে দেবেন। খেয়াল রাখুন, এটা গ্লোবাল এনভায়রনমেন্ট ভ্যারিয়েবল তখনই হয়ে উঠবে যখন এর ঠাঁই হবে গ্লোবাল কনফিগারেশন ফাইলে।

৮।। এক কুচো ব্যাশ ফাংশন

আজ দোলের সকাল, একটু ফাংশন না-করলে চলে? পরিচালকদের প্রতি প্রযোজকদের লিংগোয় বলতে গেলে, ‘এখানে ভ্যারিয়েবলটা আর একটু এক্সপোজ করা চাইছিল না?’, ঠিকই, কিন্তু, বেশি এক্সপোজ করার রিস্ক নিলাম না, বসন্তোসবের দিন বলে কথা, প্রাচীন উন্মুক্ত ভারতে যেদিন নর্মদায় যুবকযুবতীরা জন্মদিনের পোষাকে সাঁতার প্রতিযোগিতায় নামত। নর্মদার জল হুগলী অব্দি যে গড়াতে চাইবে না, কে বলতে পারে? তার চেয়ে একটু ফাংশন টাংশন করাই ভালো, লুচিশীল চল্লিশ কোমরের বাঙালির রুচিটুচিও রইল, আবার আবির মাখানোর কন্ট্রোলড বেলেল্লাও হল।

আগেই বলেছি আমরা এই ফাংশনগুলো থাকে ‘/etc/bashrc’ বা ‘~/.bashrc’ ফাইলে। গ্লোবাল ফাংশন এবং লোকাল ফাংশন। ফাংশন কী তাই নিয়ে একাধিকবার কথা হয়েছে আমাদের, কিছু নির্দিষ্ট কাজের একত্র একটা সমাহার। যে কাজগুলো প্রায়ই একসঙ্গে করতে হয়, তাদের একটা সমাহার। সমাহারটা যাতে যে কোনো প্রোগ্রাম কাজে লাগাতে পারে তার দরকার মত, তাই কিছু নির্দিষ্ট ফাংশনকে ব্যাশের কনফিগারেশন ফাইলেই দিয়ে দেওয়া হচ্ছে। ফাংশন এবং লাইব্রেরি নিয়ে আলোচনা মনে আছে? এই ব্যাশ কনফিগারেশন ফাইলটা সেই অর্থে একটা আনবিক লাইব্রেরি বলে ভাবতে পারেন।

ধরুন, আপনার লেখা প্রবন্ধগুলোকে আপনি প্রায়ই ‘bzip2’ করে ‘.bz2’ এক্সটেনশনের ফাইল করেন। করতেই হয়, কারণ, পাঠক পাওয়ার আগ্রাসী আকাঙ্খায় আপনাকে সেগুলো ইমেল করে পাঠাতে হয়, এবং একই সঙ্গে, খুবই অবিশ্বাস্য যদিও, আপনাকে খেয়ে পরে বাঁচতে হয়, এবং ‘.bz2’ মানে মূল ফাইলটার সাইজ, আরো যদি টেক্সট ফাইল হয়, মোটামুটি দশ থেকে কুড়ি শতাংশ, তার মানে আপনার অনলাইন থাকার টাইমও। আর যদি আপনি লেখেন, এই আকাঙ্খাটা কতটা বিধবংসী হতে পারে, তার প্রমাণের সিরিজে শেষতম হল সঙ্কর্ষণের মন্তব্য, ‘তুমি তো মাইরি ফিডব্যাক চেয়ে খুঁচিয়ে গ্যাংগ্রিন করে দিলে’। গত দুই দশকের আমার সংগ্রহ থেকে দুটি মাত্র সুক্তিরত্নাবলী আপনাদের শুনিয়ে রাখি। পিউ একবার বলেছিল, ‘দীপঙ্করদা, এই পরিমাণ লিখলে কিন্তু এখন থেকে শোনার চার্জ লাগবে পাতা প্রতি দশ টাকা’। আর শ্রেষ্ঠতমটা হল বিবেকের, বিবেকদংশনের মত, আমি তখন ‘তপন বিশ্বাসের খিদের বত্রিশ ঘন্টা’ বলে একটা উপন্যাস লিখছি, ’৯০-এর এপ্রিল থেকে সেপ্টেম্বর, আমার কাছে পড়তে এসে রোজ দেখত, খেয়ে না-খেয়ে লিখে যাচ্ছি, দু-একদিন একটু আধটু উল্টে পাল্টে দেখেছেও, একদিন খুব উৎসাহভরে বলল, ‘শোনো, তোমার এই উপন্যাসটা শেষ হলে আমায় একটু দিও তো, একটা লোককে হেভি বোর করার ইচ্ছে আছে’। হায়। হায় হায়।

এবার একটা ফাংশন আপনি বানিয়ে রাখতে পারেন আপনার ‘~/.bashrc’ ফাইলে যাতে এই গোটা কাজটাই এককথায় হয়ে যায়, আলাদা করে ধরে ধরে কমান্ড দিয়ে কাজটা না-করতে হয়। এই অংশটা যোগ করে দিন আপনার ‘~/.bashrc’ ফাইলে। যোগ করার পর একবার ‘<Ctrl><D>’ বা ‘`exit`’ করে বেরিয়ে যান, দেখবেন লগ-ইনের আমন্ত্রণ জ্বলজ্বল করছে কমান্ড প্রম্পটে, আর একবার লগ-ইন করে নিন। তার আগে অব্দি ব্যাশ ফাংশনটা পাবে না, কারণ নতুন ‘~/.bashrc’ সে তখনো পড়েনি। ব্যাশ ম্যানুয়াল পড়ে দেখুন, কী একটা অন্য উপায় আছে লগ-আউট না করেই কনফিগারেশন ফাইল ব্যাশকে দিয়ে পড়িয়ে নেওয়ার, আমার মনে পড়ছে না।

```
packall()
{
for file in *
do bzip2 -9 $file
done
}
```

এতে কী হল, বলুন তো? এখন থেকে যে কোনো ডিরেক্টরির মধ্যে দাঁড়িয়ে, আপনি ‘`packall`’ কমান্ড দিলেই গোটা ডিরেক্টরি জুড়ে আপনার প্রতিটা প্রবন্ধ একটা করে আলাদা আলাদা একটা ‘.bz2’ ফাইল হয়ে গেছে। পাঠান না, যত খুশি, যাকে পাঠাবেন, ঠেহায় কেডা। এবং এটা দিয়ে আপনি আপনার ডিস্কভূমিও বাঁচাতে পারেন, গোটা ডিরেক্টরির সাইজও অনেক ছোট করে দেয়, কারণ ‘`bzip2`’ করার পরে মূল ফাইলগুলো উড়িয়ে দিয়েছে। আর ‘`bzip2`’ করেছে অপশন ‘`-9`’ দিয়ে, মানে সর্বোচ্চ সম্ভব কুঁকড়ে দিয়েছে। আগের শেলস্ক্রিপ্টের ব্যাখ্যাগুলো যদি মন দিয়ে পড়ে থাকেন, তাহলে এটা বুঝতে আপনার কোনো অসুবিধে হচ্ছেনা, আর যদি না-পড়ে থাকেন, তাহলে যান দোল খেলুন। আর আপনি যদি কালেক্টিভ ফার্মিং-এ বিশ্বাসী না হন, একটা প্রবন্ধ শেষ করেন, একটা ফাইল পাঠান, তাহলে আপনার ‘~/.bashrc’ ফাইলে যোগ করবেন নিচের লাইনগুলো। এটা ‘packone()’ নামে একটা ফাংশন। আগের ‘packall()’ ফাংশনটার সঙ্গে এটার তফাত এই যে এটা আর ডিরেক্টরি ধরে কাজ করেনা, কাজ করে ফাইল ধরে। আগেরটা যেমন একটা ডিরেক্টরিতে দাঁড়িয়ে ‘`packall`’ কমান্ড দিলেই চলত, পরের ‘packone()’ ফাংশনটার জন্যে প্রথমে ‘`packone`’, তার পরে একটা ফাইলনাম যোগ করে দিয়ে কমান্ডটা সম্পূর্ণ করতে হবে। তখন ব্যাশ সেই ফাইলটাকে কুঁকড়ে একটা ‘.bz2’ ফাইল বানিয়ে দেবে। একটা জিনিষ খেয়াল করবেন প্রত্যেকটা ফাংশন হেডিং-

এ, মানে ফাংশন যেখানে তার নাম দিয়ে শুরু হচ্ছে, আমরা ফাংশনের নামের পরেই একটা জোড়া ব্রাকেট ‘ () ’ দিচ্ছি। এটাই নিয়ম। এটা দিয়েই ব্যাশ একটা ফাংশনকে চেনে। ফাংশনটা ব্যবহার করার সময় কিন্তু ব্রাকেটজোড়া আর ব্যবহার করতে হচ্ছে না।

```
packone()
{
bzip2 -9 $1
}
```

এখানে একটা জিনিষ এখনো আমাদের অজানা, ‘$1’। একটু বাদেই আমরা দেখব, এর মানে, ‘packone’ কমান্ডটার পরে যে ফাইলটার নাম আপনি দিচ্ছেন। এটা ব্যাশের আর একটা রেডিমেড ফরমুলা। ব্যাশ প্রম্পটে কোনো কমান্ড দেওয়া মানেই তার পরের শব্দটা হল ‘$1’। ধরুন, এইমাত্র আপনি শেষ করেছেন ‘essay.4.text’ ফাইলটা। আপনি এবার কমান্ড দেবেন ‘`packone essay.4.text`’। কারণ, এখন একটা ফাইলের নাম দিতে হবে। এতে ব্যাশ তার চেনা ফাংশন ‘packone’ কাজে লাগিয়ে সঙ্গে সঙ্গে ‘essay.4.text.bz2’ নামে একটা ফাইল বানিয়ে দেবে। কারণ, দেখুন, ও তুলেছে ‘essay.4.text’ ফাইলটা, তাকে ‘bzip2’ করে বানিয়েছে ‘essay.4.text.bz2’। যদি আমরা এখানে শুধু ‘`packone`’ কমান্ড দিই, কোনো ফাইলনাম না-দিয়ে, ব্যাশ আমাদের জানাবে —

```
bzip2: I won't write compressed data to a terminal.
bzip2: For help, type: `bzip2 --help'.
```

সিস্টেম বোঝার চেষ্টায় একটা খুব ভালো ব্যায়াম এই গন্ডগোলের খবর বা এরর মেসেজগুলো পড়ার এবং বোঝার চেষ্টা করা। দেখুন তো, ব্যাশ যেরকম বলেছে, ‘`bzip2 --help`’ করে বা তার বড়ভাই ‘`man bzip2`’ করে আপনি এই এরর মেসেজটা কতটা বুঝতে পারেন। এবার, ‘packone()’ আর ‘packall()’ ফাংশনদুটোর সঙ্গে একটা এর বিপরীত ফাংশনও আপনি বানিয়ে নিতে পারেন। যে কোঁকড়ানো ফাইলগুলোকে বড় করবে। ধরুন তার নাম দিলেন ‘unpack()’। তখন আপনি এই রকম কিছু লাইন যোগ করবেন আপনার ‘~/.bashrc’ ফাইলে।

```
unpack()
{
for pack in *.bz2
do bunzip2 $pack
done
}
```

এই ফাংশনটায় দেখুন ভ্যারিয়েবলটার নাম দেওয়া হয়েছে ‘pack’, আগের ‘packall()’ ফাংশনটায় যেমন দেওয়া হয়েছিল ‘file’, এটা গোটাটাই নিজের খুশি মত, ঠিক ফাংশনের নাম যেমন। এই ‘unpack()’ ফাংশনটা ঠিক ‘packall()’ ফাংশনের মত, কোনো ফাইলনাম দিতে হয়না, গোটা ডিরেক্টরির ফাইলতালিকা ধরে কাজ করে। এই যে লেখা ‘`for pack in *.bz2`’, এর মানে এই ডিরেক্টরিতে দাঁড়িয়ে ‘`ls *.bz2`’ কমান্ড দিলে আমরা যে তালিকাটা পেতাম, মানে গোটা ডিরেক্টরিতে যাবতীয় ‘.bz2’ ফাইলের তালিকা, সেই তালিকাটা পড়ে নিচ্ছে ব্যাশ। তারপর তার প্রথম ফাইলনাম হচ্ছে ‘pack’ ভ্যারিয়েবলের প্রথম মান, দ্বিতীয় ফাইলনাম হচ্ছে দ্বিতীয় মান, এই ভাবে এগিয়ে যাচ্ছে। আমাদের এই ব্যাশ ফাংশন কাণ্ড শেষ হওয়ার আগে একটা কথা মনে করিয়ে দিই, এই ফাংশনগুলো এক ভাবে দেখলে সম্পূর্ণ অর্থহীন, অ্যালিয়াস দিয়ে খুব সহজেই আপনি করে নিতে পারতেন এটা। কিন্তু, আগেই তো বলেছি, এই লেখাটা আপনাকে সাহায্য করার চেষ্টা করছে আপনি কী ভাবে নিজেকে শেখাবেন সেটা বুঝে ওঠার কাজে। তার চেয়ে বেশি এগোতে হবে আপনাকে নিজেকেই।

৯।। ব্যাশ একটা স্ক্রিপ্টিং ল্যাংগুয়েজ

বারবার নানা উদাহরণ দিয়ে নয় নম্বর দিন থেকে আমরা দেখে আসছি, ব্যাশে কমান্ড প্রম্পটে যে সব কমান্ড দিয়ে আমরা নানা কাজ করছি, সেই কমান্ডগুলোকে একসঙ্গে গেঁথে দেওয়া যায়। এইমাত্র যে ফাংশন নিয়ে আমরা আলোচনা করলাম, সেই ফাংশনও একটা গ্রথিত কমান্ডসমাহার। কিন্তু তার সঙ্গে ব্যাশ স্ক্রিপ্টের মূল পার্থক্য হল স্বাধীন স্বতন্ত্র অস্তিত্বের। ফাংশন হল একটা কমান্ডগুচ্ছ যে স্বনির্ভর এবং স্বতন্ত্র নয়, তাকে ডেকে আনে অন্য কোনো প্রোগ্রাম বা শেল। যেমন এইমাত্র আমরা দেখলাম, কমান্ড লাইন থেকে, অর্থাৎ ব্যাশ দিয়ে তাদের কী ভাবে ডাকতে

হয়। শুধু ব্যাশ কমান্ড প্রম্পট নয়, এই ফাংশনগুলোকে ডেকে নিতে পারে কোনো ব্যাশ স্ক্রিপ্টও। কিন্তু, যেই ডাকুক, তার চলার ভিতর থেকে গজিয়ে ওঠে ফাংশনটার চলা, ফাংশনটা নিজে নিজে চলেনা। তাকে নিজে নিজে চালাতে গেলে কী করতে হয় সেই কায়দাটা আমরা আগেই শিখেছি। যেমন আগের সেকশনে যে ফাংশনগুলো দেখলাম তার যে কোনো একটা, ধরুন ‘unpack()’ ফাংশনের গোটা কমান্ডমালাটা আমরা যদি একটা ফাইলে তুলে নিই, এবং তার আগে সেই ‘#!/bin/bash’ লাইনটা যোগ করে নিই, এবং গোটাটাকে একটা স্বতন্ত্র ফাইল ‘unpack.sh’ হিশেবে সেভ করে, ‘chmod’ কমান্ড দিয়ে তাকে এক্সিকিউটেবল বা চালনীয় করে নিই, তখন এই ‘unpack.sh’ ফাইলটা একটা ব্যাশ স্ক্রিপ্ট হয়ে গেল। তার কাজ হল হুবহু ওই ‘unpack()’ ফাংশনটার সঙ্গে একই, কিন্তু সে এখন স্বতন্ত্র স্বাধীন একটা ব্যাশ স্ক্রিপ্ট। সেই ফাইলটায় এই লাইনগুলো থাকতে পারে। দেখুন, এখন আমরা দুটো লাইন যোগ করেছি, এক হল এর নাম ধাম, অন্যটা এর চাকরিবাকরি। এটা একটা প্রথা, খুব কাজের প্রথা। আগেই বলেছি। আর ফাইলনামে ‘.sh’ এক্সটেনশনটাও একটা প্রথা, চলে আসছে, একটা অভ্যস্ততা। তবে যে নামই দিন ব্যাশ স্ক্রিপ্ট হিশেবে কাজ করতে তার কোনো অসুবিধে হবেনা।

```
#!/bin/bash

# Bashscript named 'unpack.sh'
# Expands all 'bz2 ' files in the directory

for pack in *.bz2
    do bunzip2 $pack
done
```

এবার এটাকে যদি কোনো ইউজার তার ‘/bin’ ডিরেক্টরিতে রেখে দেয় তাহলে সেটা ব্যাশ শেলের পথনির্দেশে চলে এল, এবং যে কোনো ডিরেক্টরি থেকেই কমান্ড প্রম্পটে এই ‘unpack.sh’ কমান্ডটা দিলেই যাবতীয় কোঁকড়ানো ফাইল বড় করার এই কাজটা করে দেবে স্ক্রিপ্টটা। শুধু ‘/bin’ ডিরেক্টরি বাদে, সেখানে কোনো ‘.bz2’ ফাইল বড় করতে চাইলে আপনাকে কমান্ড দিতে হবে ‘./unpack.sh’। ডট বা বিন্দু বা ‘.’ মানে এখানে বর্তমান বা কারেন্ট ডিরেক্টরি। গ্নু-লিনাক্সে কারেন্ট ডিরেক্টরি শেলের পথনির্দেশে থাকেনা, সিস্টেমের নিরাপত্তার সঙ্গে জড়িত তার গল্পটা বোধহয় আগেই করেছি, যদি না-করে থাকি নিজে খুঁজে পড়ে নিন।

ব্যাশ শেল তাই শুধু একটা কমান্ড প্রম্পট না, নিজেই একটা প্রোগ্রাম করার ভাষা, যাকে বলে স্ক্রিপ্টিং ল্যাংগুয়েজ। একে কিন্তু প্রোগ্রামিং ল্যাংগুয়েজ বলেনা, বলে স্ক্রিপ্টিং, খেয়াল করুন। সি ইত্যাদি প্রোগ্রামিং ভাষায় লেখা কোড চালাতে গেলে আগে কম্পাইল করে নিতে হয় কোডটা। গ্নু-লিনাক্সের কম্পাইলার জিসিসি, গোটা গ্নু-লিনাক্স সিস্টেমটাই যার উপর দাঁড়িয়ে আছে, সেটা আপনার গ্নু-লিনাক্স মেশিনে এমনিতেই আছে। সেই কম্পাইলার দিয়ে আপনি যখন সি ভাষায় লেখা প্রোগ্রাম কম্পাইল করেন, জিসিসি সেই মানববোধ্য কোডটাকে পড়ে মেশিনবোধ্য চালনীয় ফাইল বা এক্সিকিউটেবল ফাইল করে দেয়, আগেই বহুবার বলেছি। এই গোটা কাজটাকেই চালু অর্থে প্রোগ্রামিং বলা হয়। সি বা অন্য যে কোনো হাইলেভেল মানববোধ্য ভাষার নিয়মকানুন মেনে কিছু কোড লিখলেন, কোডিং, তাকে কম্পাইল করলেন, কম্পাইলিং, কম্পাইল করতে গিয়ে যে যে জায়গায় আটকাল বা পরে চলার সময়ে বুঝলেন যে যে জায়গায় ত্রুটি রয়ে গেছে, বা আর একটু ভালো করে করা যেত — সেগুলো সংশোধন করলেন বা পোকা বাছলেন, ডিবাগিং, এবং সবশেষে পোকা ছাড়ানো মনোমত কোড থেকে কম্পাইল করে চূড়ান্ত চালনীয় ফাইল বা এক্সিকিউটেবল ফাইল বানালেন — এই গোটাটাকে মিলিয়েই হল প্রোগ্রামিং।

এর সঙ্গে স্ক্রিপ্টিং-এর কিছু মৌলিক তফাত আছে। স্ক্রিপ্টিং বলতে আমরা বোঝাচ্ছি, কিছু ব্যাশ বা অন্য কোনো শেলের কমান্ডকে নেওয়া, তাদের একত্রে সমাহৃত করা একটা ফাইলে, স্ক্রিপ্ট লেখার নিয়মকানুন মেনে, যেমন গোড়াতেই ‘#!/bin/bash’ ইত্যাদি ওই বিশেষ লাইনটা লাগিয়ে নেওয়া একটা প্রাথমিকতম নিয়ম, আর ‘chmod’ দিয়ে এক্সিকিউটেবল করে নেওয়া আর একটা, তারপর তাকে চালানোর চেষ্টা করা এবং যথারীতি পোকা ছাড়ানো, আর সব শেষে চূড়ান্ত ফাইলটা চালানো — একে বলে স্ক্রিপ্টিং। এখানে একটা মজা আছে, খেয়াল করুন, আমরা তো আমাদের এক নম্বর দিনেই কম্পিউটারকে, একটু আদর করেই, গাধা বলে ডেকেছিলাম, এবং সেই সূত্রে বলেছিলাম, মেশিন জিনিষটা আসলে একটা দামী গাম্বাট। অত্যন্ত নিরেট। নিজে কিছুই বোঝেনা। তাকে প্রতিটি আদেশ তার মত

করে বুঝিয়ে দিতে হয়। কম্পাইলারটা আসলে যা করে। আমাদের প্রখর বুদ্ধি আর মেশিনের জটিল এবং মহার্ঘ নির্বুদ্ধিতার মধ্যে দোভাষির কাজ করে। কিন্তু স্ক্রিপ্টের বেলায় তো কোনো কম্পাইলার লাগছে না, আমাদের চেনা পুরোনো মেশিনটাকে এমন বৈপ্লবিক ব্রেনোলিয়া খাওয়াল কে? আর কে? সেই ব্যাশ। ব্যাশই এখানে দোভাষি। কিন্তু তার অনুবাদ-কাঠামোটা কম্পাইলারের অনুবাদ প্রক্রিয়া থেকে আলাদা। একে বলে ইন্টারপ্রিটেশন, এবং যে প্রোগ্রাম এই অন্যরকম অনুবাদ মানে ইন্টারপ্রিটেশনের কাজ করে তাকে বলে ইন্টারপ্রিটার। এখানে ব্যাশ হল সেই ইন্টারপ্রিটার।

প্রোগ্রামিং ভাষা আর স্ক্রিপ্টিং ল্যাংগুয়েজ এর মধ্যে প্রাথমিক পার্থক্যটা এই যে প্রোগ্রামিং ভাষা স্বভাবত অনেক বেশি শক্তিশালী। এবং কাজ করতে পারে অনেক দ্রুত। প্রোগ্রামিং ভাষার উদাহরণ সি, ফোর্ট্রান, সিপ্লাসপ্লাস, জাভা ইত্যাদি। এই ভাষাগুলোয় লেখা সোর্স কোড কম্পাইল করে বাইনারি বানানো এবং একটা অপারেটিং সিস্টেম থেকে অন্য অপারেটিং সিস্টেমে তাদের স্থানান্তরযোগ্যতা বা পোর্টেবিলিটি নিয়ে আমরা আলোচনা করেছি চার পাঁচ ছয় এবং আট নম্বর দিনে। একই সোর্স কোড থেকে একটা অপারেটিং সিস্টেমের জন্যে বানানো এক্সিকিউটেবল অন্য অপারেটিং সিস্টেমে চলে না। আবার ওই অপারেটিং সিস্টেমের মত করে তাকে কম্পাইল করে নিতে হয়।

একটা স্ক্রিপ্টিং ভাষাও সেই অর্থে শুরু করে সোর্স কোডে। আমাদের বানানো স্ক্রিপ্টগুলোয় আমরা যে আদেশ বা কমান্ড সমাহার টাইপ করে তুলে দিচ্ছি সেটাই ওই স্ক্রিপ্টের সোর্স কোড। কম্পাইলার গোটা সোর্স কোডকে একত্রে চটকায় মানে প্রসেস করে। কিন্তু ইন্টারপ্রিটার হল শোভন রুচিশীল, সে আদেশ বাই আদেশ এগোয়। একটা ধরো পড়ো করো, তারপর আর একটা। কিন্তু এই রকম ভেঙে ভেঙে পড়ার কারণে তার গতিটাও কম্পাইলড ভাষাগুলোর তুলনায় অনেক কমে যায়। কিন্তু সুবিধে আছে এই যে এই কমান্ড গুলোকে বুঝতে পারে এমন কোনো শেল যদি একটা অপারেটিং সিস্টেমে থাকে, সেই অপারেটিং সিস্টেমেই এই স্ক্রিপ্টগুলো চলবে। যেমন ধরুন আজকের আলোচনার এক নম্বর সেকশনেই আমরা 'mplayer' নামে প্যাকেজের সোর্স কোড থেকে বাইনারি কম্পাইল করে নেওয়ার কথা বলেছিলাম। প্রথমে '`configure`' কমান্ড দিয়েছিলাম। এই 'configure' একটা প্রোগ্রাম যা গোটা মেশিনের অপারেটিং সিস্টেমের কনফিগারেশনের খুঁটিনাটি দেখে নেয় এবং সেই মোতাবেক একটা ফাইল তৈরি করে যা পরে জিসিসি নামে কম্পাইলারের কাজে লাগবে কম্পাইল করার সময়। ওই যে '`make`' এবং '`make install`' কমান্ড দুটো — ওরাও ডেকে এনেছিল জিসিসি নামের কম্পাইলারকে, ঠিক কী কী ভাবে কম্পাইল এবং ইনস্টল করতে হবে এইমাত্র কনফিগার করা তথ্য মেনে — সেই কথা বলে দিয়েছিল। 'make' হল গ্নু-র একটা উপযোগিতা প্যাকেজ, রিচার্ড স্টলম্যান এবং রোলাঁ ম্যাকগ্রার বানানো। একটু মেক-এর খেজুরগাছে চড়ে আসতে পারেন, তাতে আপনার উপকারই হবে। আর ভীষণ ভালো একটা বই, 'লিনাক্স ডিভেলপমেন্ট প্ল্যাটফর্ম', রফিক উর রহমান এবং ক্রিস্টফার পলের লেখা, মোটের উপর গ্নু-লিনাক্স সিস্টেমে যে কোনো প্রোগ্রামিং করা নিয়ে, নেটে ফ্রিতে ডাউনলোড করা যায়, সেটা পড়ে নিতে পারেন, কোনো রকমের প্রোগ্রামিং নিয়ে আপনার যদি মিনিমাম আগ্রহ থাকে। তাতে মেক নিয়ে বিরাট একটা চ্যাপ্টার আছে। মেক কাজ করে মেকফাইল (makefile) দিয়ে, যে মেকফাইল হল ওই এমপ্লেয়ার কোড ডিরেক্টরিতে রাখা আছে — একটা গাইড, কী ভাবে কম্পাইল করতে হবে। সেই ফাইল অনুযায়ী জিসিসি কম্পাইল করে একটা বাইনারি বানায়, যার নাম 'mplayer', এবং তাকে '/usr/local/bin' ডিরেক্টরিতে রেখে দেয়। এই ডিরেক্টরি ব্যাশ শেলের স্বাভাবিক পথনির্দেশের মধ্যে পড়ে, তাই এর পর থেকে, যে ডিরেক্টরিতে বসেই আমি '`mplayer`' কমান্ড দিই-না কেন, আমি সিনেমা দেখতে পাব বা গান শুনতে পাব। এই গোটা কাজটাই আমি করেছিলাম আমার মেশিনের সুজে ৮.২ সিস্টেমে। তার মানে এর অন্তর্বর্তী কনফিগারেশনটা এই সিস্টেমের। এবার এটা লিখতে লিখতেই শখ হল, কী হয় দেখি, আমি সুজে সিস্টেমের ওই '/usr/local/bin' ডিরেক্টরি থেকে 'mplayer' বাইনারিটা কপি করে নিলাম স্ল্যাকওয়ার সিস্টেমে। এবং স্ল্যাকওয়ার সিস্টেমে বুট করে তাকে চালাতে গেলাম। চলল না, প্রমাদ-বার্তা মানে এরর মেসেজ দিল।

```
mplayer: error while loading shared libraries: libdv.so.2:
cannot open shared object file: No such file or directory
```

অথচ, স্ল্যাকওয়ারে যখন কম্পাইল করে নিলাম, স্ল্যাকওয়ার সিস্টেমে আবার তার নিজের '/usr/local/bin' ডিরেক্টরিতে একই রকম একটা 'mplayer' বলে বাইনারি তৈরি হয়ে গেল। এবং সে একদম স্বাভাবিক ভাবেই চলল।

এখানে আমি আপনাদের দুটো প্রশ্ন করব, যার একটারও উত্তর দেবনা। এক, এই প্রমাদবার্তাটার আগামাথা কোনো আন্দাজ করতে পারছেন কি? আর দুই, স্ল্যাকওয়ার সিস্টেমে যেটা '/usr/local/bin' ডিরেক্টরি সেটা আমার সুজে সিস্টেমে কী ডিরেক্টরি হবে? মনে করুন, আগেই বলেছি, আমার স্ল্যাকওয়ার পার্টিশনটা সুজে সিস্টেমে বুট হয় '/mnt/slackware' ডিরেক্টরিতে। অর্থাৎ, প্রোগ্রামিং ভাষায় সি-তে লেখা 'mplayer' প্যাকেজের সোর্স কোড কম্পাইল করে একটা সিস্টেমে বানানো বাইনারি অন্য সিস্টেমে চলে-না, যেমনটা ঘটার কথা। অথচ ব্যাশ স্ক্রিপ্টগুলো খুব আরামসে চলে এমন যে কোনো সিস্টেমেই যেখানে ব্যাশ আছে। নিজে চালিয়ে দেখুন। ব্যাশ স্ক্রিপ্টিং ভাষায় সচরাচর খুব জাম্বো সাইজের প্রোগ্রাম করা হয়না, তার জন্যে আছে পার্ল (Perl), লিস্প (Lisp), টিসিএল (Tcl) ইত্যাদি।

আমাদের এক নম্বর দিনে আমরা একটা কুচো সি প্রোগ্রাম তুলে দিয়েছিলাম, সেই প্রোগ্রামটা আর একবার তুলে দিই এখানে, তারপর সেই প্রোগ্রামটাই আমরা ব্যাশ দিয়ে করব। এটা একটা চালু প্রথা যে কোনো প্রোগ্রামিং-এর আলোচনার, এই ধরনের একটা কথা-ফুটিয়ে-তোলা প্রোগ্রাম দিয়ে শুরু করা।

```
#include <stdio.h>
int main(void)
{
    printf("\nKire Gadha!!!\n");
    return 0;
}
```

এই কোডটাকে, যেমন এক নম্বর দিনে আমরা বলেছিলাম, আপনি 'gadha.c' নামে সেভ করতে পারেন। নামের শেষে ওই '.c' এক্সটেনশনটা হল সি-প্রোগ্রাম হিশেবে চিনে নেওয়ার চালু প্রথা। এবার এটাকে কম্পাইল করে নিতে পারেন জিসিসি (gcc) দিয়ে। তার জন্যে আলাদা করে কিছু ইনস্টলও করতে হবেনা। জিসিসি কম্পাইলার আপনার গ্নু-লিনাক্স সিস্টেমে গোড়া থেকেই ইনস্টল হয়ে আছে। যে ডিরেক্টরিতে আপনি 'gadha.c' ফাইলটা সেভ করেছেন সেই ডিরেক্টরিতে দাঁড়িয়ে আপনি যদি কমান্ড দেন '`gcc -o gadha gadha.c`', তাহলে জিসিসি ওই ডিরেক্টরিতে 'gadha' নামে একটা বাইনারি ফাইল বানিয়ে দেবে। এই '`-o`' অপশনটা দিয়ে আপনি বাইনারিটা যে 'gadha' নামে বানাতে হবে সেটা জিসিসিকে দিয়ে দিলেন। যদি আপনি না দেন, তাহলে জিসিসি বা গ্নু-কম্পাইলার-কালেকশন বাইনারিটা বানাবে 'a.out' নামে। এবং পরের বার আর একটা কোড ফাইল কম্পাইল করলে তার 'a.out' ফাইলটা সেভ করে দেবে এই 'a.out' ফাইলের জায়গায়। এই নামটা কেন 'a.out' হয় তার আবার একটা গল্প আছে — যাকগে, সে অন্য কথা। এবার এই বাইনারিটাকে চালাতে হবে, এই ডিরেক্টরিতে দাঁড়িয়ে চালাতে গেলে কী কমান্ড দিতে হবে সেটা আমরা জানি, '`./gadha`'। কমান্ড প্রম্পটে এই আদেশটা দিয়ে এন্টার মারলেই স্ক্রিনে লেখা ফুটে উঠবে, উপরে এক লাইন ফাঁকা জায়গা রেখে।

```
Kire Gadha!!!
```

এবার ধরুন এই গোটা কাজটাই আমরা সি নামের প্রোগ্রামিং ল্যাংগুয়েজ দিয়ে না-করে ব্যাশ নামের স্ক্রিপ্টিং ল্যাংগুয়েজ দিয়ে করতে চাইছি। তখন আমাদের একটা ব্যাশ স্ক্রিপ্ট বানাতে হবে। একটা ফাইলে নিচের লাইনদুটো ঢোকালাম আমরা।

```
#!/bin/bash
echo -e "\nKire Gadha"
```

এবং এই ফাইলটাকে সেভ করে দিলাম 'gadha.sh' নামে। এবার পরের স্টেপদুটোও আমাদের পরিচিত। এক নম্বর, এক্সিকিউটেবল করা। নানা ভাবে করা যায়। তার একটা হল '`chmod +x gadha.sh`'। বা, সংখ্যা দিয়ে করতে চাইলে, '`chmod 700 gadha.sh`'। নিজে 'chmod'-এর ম্যানুয়াল পড়ে মিলিয়ে নিন, আগেও আমরা কথা বলেছি এটা নিয়ে, সাত নম্বর দিনে।এবার দু-নম্বর স্টেপ হল এটা চালানো, মানে '`./gadha.sh`'। এবারেও ফলাফল সেই একই। অন্তত লাইনের নিরিখে স্ক্রিপ্টটা বহু ছোট, আর কম্পাইল করার কোনো ঝামেলা নেই। এবং যে কোনো অপারেটিং সিস্টেমে চলবে, সেখানে ব্যাশ থাকলেই। সি-তে কম্পাইল করা 'gadha' বাইনারিটা তা নয়, প্রতিটি আলাদা অপারেটিং সিস্টেমের জন্যে তাকে আলাদা করে কম্পাইল করে নিতে হবে। প্রোগ্রামিং জগতের সমস্ত

সমস্যা যদি এই মেশিনকে ‘কিরে গাধা!!!’ ডাকার কাছাকাছি সরলতার হত, তাহলে যে ব্যাশ স্ক্রিপ্ট দিয়েই আমাদের যাবতীয় কাজ চলে যেত, অন্য কোনো প্রোগ্রাম করার মাধ্যম তৈরিই হত না, এতে কোনো সন্দেহ নেই। জটিলতার নিরিখে প্রোগ্রামিং ল্যাংগুয়েজরা ব্যাশের থেকে অনেকটা এগিয়ে থাকলেও ব্যাশের নিজের কিছু জোরের জায়গা থাকেই। এক, আলাদা করে চালাতে হয়না। আপনার গ্নু-লিনাক্স চলছে মানে তার উদার উদরের ভিতরে ব্যাশও চলছে, এমনিতেই। এবার, নতুন করে কোনো ব্যাশ স্ক্রিপ্ট চালানো মানে, একটা নতুন ব্যাশ প্রক্রিয়া বা প্রসেস, যা অনেকটা অব্দি মেমরি তথা অন্যান্য রসদ বাঁটোয়ারা করে নেবে, শেয়ার করবে ইতিমধ্যেই বাফারে থাকা ওই প্রারম্ভিক ব্যাশ প্রক্রিয়ার সঙ্গে। কোনো কম্পাইলার, বা রুবি বা পার্ল ইত্যাদি অন্য কোনো ইন্টারপ্রিটার মানেও সেটা তো কিছুটা আলাদা মেমরি নেবেই। ব্যাশের দুই নম্বর প্লাসপয়েন্ট এই যে ব্যাশ আমাদের এমনিতেই চেনা জিনিষ। সিস্টেমের মধ্যে কাজ করা মানেই ব্যাশ শিখতে থাকা। পাঁচ নম্বর দিন থেকে এই অব্দি আমরা যা যা কমান্ড লাগিয়ে সিস্টেমে যা যা করেছি, তা সবই তো ব্যাশ। জ্যান্ত রকমে ইন্টারাকটিভলি ব্যাশকে ব্যবহার করে আসছি, আলাদা আলাদা কেজো কমান্ডের এককে। সে ফাইল কপি করা হোক, নড়ানো হোক, বা পাইপ করা রিডাইরেক্ট করা ইত্যাদি কিছু হোক। এবার, ব্যাশকে যখন ইন্টারাকটিভ রকমে না-ভেবে, একটা স্ক্রিপ্টিং ল্যাংগুয়েজ হিশেবে ভাবছি, এই কমান্ডগুলো এই আকারেই থাকছে, শুধু তাদের একত্রে আনার আলাদা কিছু নিয়ম চালু হচ্ছে মাত্র। তাই ব্যাশ আলাদা করে শেখা শুরু করতে হয়না। ম্যাক্সিমাম কিছু বাড়তি খুঁটিনাটি জেনে নিতে হয়, যখন যা দরকার। তিন, ব্যাশের গোটা ডকুমেন্টেশনটাই ইতিমধ্যেই সদাসর্বদাই রয়েছে আপনার সিস্টেমে। ‘`man bash`’ বা ‘`info bash`’ করুন এবং পড়ে যান। আর পাশের কনসোলে সেগুলো করে করে দেখতে থাকুন। এক্সউইনডোজে থাকলে, আর একটা টার্মিনাল উইনডোয়।

এই যে ব্যাশ স্ক্রিপ্টটা আমরা এইমাত্র চালালাম, ‘gadha.sh’ — এটা আসলে কী? নিছক একটা কমান্ড, ‘echo’ নামের প্রোগ্রামটা ব্যবহার করে, যাকে কমান্ড প্রম্পটে টাইপ করে দিয়ে এন্টার মারলে হুবহু একই কাজ করত। তার মানে আমাদের স্ক্রিপ্টটা আর কিছু করেনি, নিছক একটা সিস্টেমে থাকা কমান্ডকে ব্যবহার করেছে। ব্যাশ কোড ঠিক এটাই করে, পরপর সিস্টেমে থাকা এক বা একাধিক কমান্ডকে তাদের অপশন সহ বিভিন্ন ভ্যারিয়েবলের উপর প্রয়োগ করে যায়। নয় নম্বর দিনে ‘writefiles’ নামের ব্যাশ স্ক্রিপ্টটা আমরা লাইন ধরে ধরে আলোচনা করেছিলাম। স্ক্রিপ্টের লাইনকটা মনে করুন।

```
#!/bin/bash
echo Name of file?
read f
c=1
d=**********************************************
for i in *
    do echo -e "\n\n$d\n$c. $i\n" >> $f
    cat $i>> $f
    c=`expr $c + 1`
done
```

এই স্ক্রিপ্টটা ‘gadha.sh’-এর চেয়ে একটু বড়, ব্যবহার করেছে চারটে কমান্ড ‘echo’, ‘read’, ‘cat’ এবং ‘expr’ তাদের অপশন সহ। এবং এই কাজগুলো করে চারটে ভ্যারিয়েবলের উপর, লিখবে যে ফাইলে তার নাম মানে ‘f’, কাউন্টার মানে ‘c’, বিভাজক বা ডিভাইডর মানে ‘d’, এবং ডিরেক্টরির ফাইলগুলো একের পর এক মানে ‘i’। আর এই সমস্ত আদেশগুলোকে সঠিক বোধ্য রকমে ব্যাশের কাছে পৌঁছে দেওয়ার জন্যে আমাদের লেগেছে কিছু বিশেষ যতি চিহ্ন। যেমন, মন্তব্যবাচক চিহ্ন ‘#’, বা চালান করা মানে রিডাইরেকশন চিহ্ন ‘>>’, বা কমান্ড সাবস্টিটিউশনের চিহ্ন ব্যাককোট ‘``’, বা অ্যাসাইনমেন্ট চিহ্ন ‘=’, বা ভ্যারিয়েবলের মানবাচক চিহ্ন ‘$’।

এই গোটাটাকে মিলিয়ে হল ব্যাশ নামের স্ক্রিপ্টিং ল্যাংগুয়েজ। তার কমান্ডগুলো কারা? অন্য প্রোগ্রামিং ভাষায় যেমন কমান্ডগুলোকে আলাদা করে শিখতে হয়, এখানে তা নয়, প্রতিমুহূর্তে একটা গ্নু-লিনাক্স সিস্টেমে কাজ করতে গিয়ে আমরা যে কমান্ডগুলো আদেশগুলো ব্যবহার করে আসছি সেই পাঁচ নম্বর দিন থেকে, সেই কমান্ড গুলোই একদিকে যেমন ব্যাশ নামক কমান্ড প্রম্পটের কমান্ড, অন্যদিকে তেমনি ব্যাশ নামক ল্যাংগুয়েজেরও কমান্ড। আর ভ্যারিয়েবল গুলো নিয়ে এবং দু একটা চিহ্ন নিয়ে আমরা তো আলোচনা একটু আগেই সেরে এলাম। এবার আমরা ব্যাশ শেলের

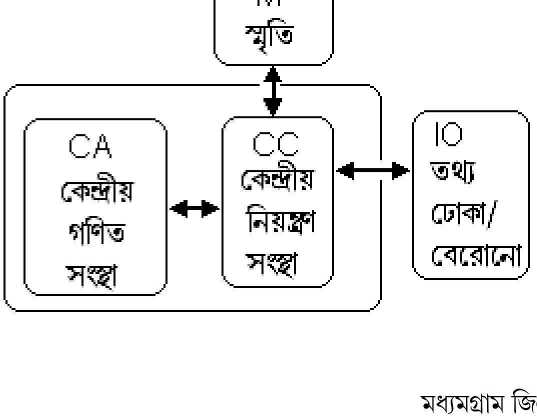
ভাষাগত এবং ক্রিয়াগত নিয়মকানুনগুলো আর একটু খুঁটিয়ে শিখব। যাকে আমরা বলব কন্ট্রোল স্ট্রাকচার বা নিয়ন্ত্রণ কাঠামো। মনে করে দেখুন, শূন্য নম্বর দিনের সেই আলোচনা, যা আবার আডার কথায় ফেরত এসেছিল, তিন নম্বর দিনে, ক্যালকুলেটর থেকে কী সেই নাটকীয় উল্লম্ফন যা কম্পিউটারকে কম্পিউটার করল? সেটা এই নিয়ন্ত্রণ কাঠামো।

১০।। ভন নয়মান আর্কিটেকচার এবং নিয়ন্ত্রণ কাঠামো

শূন্য নম্বর দিনে আমরা ভন নয়মানের উল্লেখ করিনি ইচ্ছে করেই। স্টিফেন হকিংয়ের ব্রিফ হিস্ট্রি অফ টাইমের ভূমিকাতেই ছিলনা সেই প্রসঙ্গটা, গাণিতিক সমীকরণের সংখ্যার বর্গের ব্যস্তানুপাতে কমে যেতে থাকে পাঠকের সংখ্যা? খটোমটো এবং অপরিচিত নামের বেলাতেও একই ভাবে সেটা সত্যি। আর সঠিক জায়গায় সঠিক নাম ত্যাগ করে (মলত্যাগের সঙ্গে যদি কোনো আত্মীয়তা পান, সেটা আমার বানানো নয়, ইংরিজিতে বলে নেম-ড্রপিং), সঠিক লোকের কাছে পৌঁছে দেওয়া, লাইনেই আছি দাদা, আত্মবিশ্বাসহীনতার এই কলোনিয়াল সংস্কৃতিটারও বাইরে যেতে চাইছিলাম। সাত নম্বর দিনে খুব উত্তেজিত ভাবে লিখেছিলাম না, নেম-ড্রপিং সিম্বল-ড্রপিং করে আবহটাকে ইচ্ছে করে আরো খটোমটো আরো এলিট করে দেওয়া এটাও একটা রাজনীতি। আমাদের ঘরের কোনো টুম্পা যাতে আডা লাভলেস হওয়ার স্পর্ধা করার স্বপ্নেও পৌঁছতে না-পারে তার রাজনীতি। যাতে লিনাক্স এইট করা লিনাক্স বিশারদ ধেড়ে পাঠারা সাহেব অ্যাকাডেমির সাহেব ড্রপিং ঘেঁটে যেতে পারে, আর তার সৌরভে আমোদিত হতে পারে বন্দুকের আওয়াজে বমকে যাওয়া স্যাভেজের বাচ্চারা — দশদিক থেকে যারা আসিতেছে চলে পিসি পিসি শিখবে বলে। মার্জিন অফ মার্জিন বলে আমাদের একটা বইয়ে একটা শব্দ আছে ‘savage’, যাকে বানানে লেখা ‘sa**VA**ge’। প্রথম আর শেষটুকু মিলিয়ে হয় সেজ, সাধক, তপস্বী, প্রাজ্ঞ। আমি ব্যক্তিগতভাবে একজন হতভাগ্য পরিণামহীন ভবিষ্যতহীন মানুষ, একদম জৈবনিক অর্থেই। আজকাল মনে হয়, আমাদের সমস্ত সমাজ বদলানোর স্বপ্নই আত্মহত হয়েছে চালু ইতিহাসের হাতে, নিজেকে নিজের জীবনকে নিয়ে তাই আর কিছু এসে যায়না, নাউ দ্যাট আই অ্যাম নাইন্টিথ্রি, আই কেয়ার এ ড্যাম ইউ সি, কিন্তু একটা টুম্পাও যদি আডা হতে চেষ্টা করার উত্তেজনাটাও পেতে পারে — তার চেয়ে বড় পাওনা আর কী হয়? একটা স্যাভেজও যদি এই নিজের মধ্যে সঙ্গোপন সেজ খোঁজার প্রক্রিয়ায় পৌঁছতে পারে, তার একটুও রসদ পায় আমার কাজ থেকে। এই গোটা বইটায় প্রতিমুহূর্তে আপনাকে এগিয়ে দেওয়ার চেষ্টা করা আমাদের অ্যাকাডেমির ধেড়ে পাঠাদের সেই মহিমার রাজনীতির একটা বিপরীত রাজনৈতিক চেষ্টার একটা ব্যক্তিগত যাত্রা। সেই যাত্রায় সঙ্গীও তো জুটেছে এই গ্নু-লিনাক্স জগতের থেকেই, অন্তত পাঁচটা লোক সত্যিই উত্তেজিত এই বইটা ঘিরে — সঙ্কর্ষণ, অশেষ, তথাগত, পিউ আর দেবাশিস। আরো অনেক বেশি জীবনীশক্তি নিয়ে আরো অনেক জীবন্ত রকমে যখন সমাজ বদলানোর স্বপ্ন দেখেছিলাম, তখন তো এটুকু উত্তেজনাও তৈরি করতে পারিনি নিজের চারপাশের রাজনীতিতে।

ফেরত আসি সেই নাম না-করা ভন নয়মান আর্কিটেকচারের কথায়। আমরা কাজের ভিত্তিতে একটা কম্পিউটারের পাঁচ রকমের উপাদানের কথা বলেছিলাম। এক, তথ্য ঢোকানো বা ইনপুট ডিভাইস। দুই, তথ্য বার করা বা আউটপুট ডিভাইস। তিন, তথ্য চটকানো বা প্রসেসিং ডিভাইস। চার, তথ্য ধারণ বা হোল্ডিং ডিভাইস। আর পাঁচ, সবচেয়ে গুরুত্বপূর্ণ, এক থেকে চার এই চারটে কাজের নিয়ন্ত্রণ বা কন্ট্রোল ডিভাইস। এবার এই ছবির সঙ্গে মিলিয়ে ভন নয়মানের মূল ছকটা দেখুন। শুধু একটা জিনিষ, এখানে ইনপুট আর আউটপুটের উপাদানগুলোকে তাদের কাজের ধরনের আত্মীয়তার ভিত্তিতে একটা উপাদান হিশেবে ধরা হয়েছে।

ক্রিয়ার যুক্তিতে কম্পিউটারের গঠন
ভন নয়মান আর্কিটেকচার



কম্পিউটার বিজ্ঞানে ভন নয়মানের অবদানের কথা আমরা আগেই উল্লেখ করেছি, চার নম্বর দিনে। এখানে আমরা শুধু পেনসিলভানিয়া বিশ্ববিদ্যালয়ে থাকাকালীন এডভ্যাক (EDVAC) কম্পিউটারের সূত্রে ১৯৪৫ সাল নাগাদ তিনি যে ছকটা দিয়েছিলেন স্টোরড-প্রোগ্রাম-কম্পিউটারের লজিকাল গঠনের, সেটাকেই তুলে দিচ্ছি। স্টোরড-প্রোগ্রাম বিষয়টা খেয়াল করুন, এর মানে যেখানে গোটা প্রোগ্রাম বা আদেশ-

তথ্যটাকেই স্মৃতিতে তুলে কম্পিউটার কাজ করতে পারে। কম্পিউটারকে এখানে মূল চারটে অংশে ভাগ করা হয়েছে — সেন্ট্রাল অ্যারিথমেটিকাল ইউনিট বা কেন্দ্রীয় গণিত সংস্থা (CA), সেন্ট্রাল কন্ট্রোল ইউনিট বা কেন্দ্রীয় নিয়ন্ত্রণ সংস্থা (CC), মেমরি বা স্মৃতি (M) এবং ইনপুট/আউটপুট বা তথ্য ঢোকা/বেরোনোর সংস্থা (IO)। 'CA' করবে গাণিতিক হিশেবের কাজ। চারটে মূল পাটীগাণিতিক ক্রিয়া, মানে যোগ বিয়োগ গুণ ভাগ, এবং তাদেরকে একত্রে কাজে লাগিয়ে সূচক বা মূলের, মানে ইনডেক্স বা রুটের, তথা লগ এবং ত্রিকোনমিতিক ফাংশনের বা অপেক্ষকের হিশেবের কাজ। 'M' ধরে রাখবে তথ্য, গাণিতিক তথ্য মানে বিভিন্ন সংখ্যা, যাদের নিয়ে সে কাজ করবে, এবং কী ভাবে কী কাজ সে করবে তার নির্দেশের গোটা তথ্যটাও তোলা থাকবে এই 'M'-এ। এই নির্দেশ তথ্যকে এখন আমরা চিনি — বাইনারির ভিতরের কম্পাইলড কোড বা ইন্টারপ্রিট করা স্ক্রিপ্টেড কোড। আর 'IO' মানে হল ব্যবহারকারীর সঙ্গে কম্পিউটারের পারস্পরিক সম্পর্কের ইন্টারফেস, কাঁচামাল তথ্য ঢোকানো এবং উৎপাদিত তথ্য বার করার। আর 'CC' করবে অন্য কাজ গুলোকে নিয়ন্ত্রণের কাজ। কোন তথ্যের উপর কোন নির্দেশ কাজ করবে, কতদূর অব্দি কী কাজ শুরু করবে, কখন কোন কাজ শুরু বা শেষ হবে, কখন কোন কাজ বদলাবে, ইত্যাদি।

ভন নয়মানের এই ছকটা একটা লজিকাল বা যুক্তিবৈজ্ঞানিক ছক। মেশিন বা যন্ত্রাংশের ভৌত গঠন নিয়ে মাথা ঘামাননি নয়মান। কম্পিউটার ক্রিয়ার যুক্তিগত প্রবাহটাকে ধরতে চেয়েছিলেন। এবার একটা অন্যরকম ক্রিয়াকে ভাবুন। ধরুন আপনি কিছু হিশেব কষছেন ক্যালকুলেটর নিয়ে। তখন যে কাগজে রাফ হিশেবগুলো লিখছেন সেটা নিশ্চিতভাবেই 'M'। ক্যালকুলেটরের বোতামগুলো আর ডিসপ্লেটা মিলিয়ে তার 'IO'। ক্যালকুলেটর তথা তার অভ্যন্তরে সার্কিটটা হল 'CA'। কিন্তু 'CC' কই? 'CC' কী নয়, প্রশ্নটা এখানে হবে, 'CC' কে? কর্ম নয়, কর্তা। এখানে আপনি হলেন 'CC'। সচেতন নিয়ন্ত্রক সংস্থা। ক্যালকুলেটরটা ক্যালকুলেটর বলেই একটা সচেতন নিয়ন্ত্রক সংস্থার প্রয়োজন পড়ছে। আর স্টোরড-প্রোগ্রাম-কম্পিউটারের আপনি সেই নিয়ন্ত্রণটাকে বকলমা দিয়ে দিচ্ছেন, পাওয়ার-অফ-অ্যাটর্নি করে দিচ্ছেন কম্পিউটারটাকে। প্রোগ্রামিং ল্যাংগুয়েজ-এর বেলায় কোড এবং কম্পাইলার এবং বাইনারির যৌথ হাতে। আর ব্যাশ স্ক্রিপ্টিং-এর বেলায় ব্যাশ স্ক্রিপ্ট এবং ব্যাশের যৌথ হাতে। এরাই হয়ে দাঁড়াচ্ছে ওই সচেতন নিয়ন্ত্রণের প্রতিনিধি। আর নিয়ন্ত্রণের ভৌত ক্রিয়াটা ঠিক কার হাতে থাকছে সেটা যদি আপনি মাথায় আনতে না-পারেন, তাহলে বড্ড ভুল জায়গায় পড়ছেন, আপনার ফের শূন্য থেকে শুরু করার কথা।

এবং এবার আমাদের ব্যাশ নামক স্ক্রিপ্টিং ভাষাটার ব্যাকরণটা একটু জানতে হবে, জ্যান্ত মানুষের সচেতন নিয়ন্ত্রণটাকে সে ঠিক কী ভাবে বাস্তবে প্রয়োগ করে সেটা জানতে।

১১।। চিহ্নপ্রবাহ এবং তার দিকনির্দেশ

মাউন্ট ব্যাটন সাহেবঅ, তোমার সাধের ব্যাটন কার হাতে থুইয়া গেলাঅ — আজাদিপ্রাপ্তির পর চারদিকের ব্যাপার স্যাপার দেখার বেজায় বেদনা ছিল হেমাঙ্গ বিশ্বাসের গানে, আমাদের এরকম আক্ষেপের কোনো কারণ নেই। আমাদের ব্যাটন আমরা সাভিলাষ সারাম দিয়ে দিয়েছি ব্যাশকরপুটে। ব্যাশ এবার গোটাটা দেখভাল করছে, মেশিনকে ঘিরে গোটা ক্রিয়াকলাপের। এই ক্রিয়াকলাপটা আসলে কী? সেই শূন্য এক দুই তিন করে যে দিনগুলো আমরা পেরিয়ে এলাম, তার গোটাটা মাথায় রেখে একবার ভাবুন তো? শূন্যয় সেই বলেছিলাম না, শেষ অব্দি একটা কম্পিউটারের ক্রিয়াকলাপ খুব সরল সাধাসিধে, ডিরেক্ট। কোনো ধূসরতার সন্ধ্যাভাষার ধুন্ধলে শাহি উপাখ্যান সেখানে ছুপা নেই। গোটাটাই বাইনারি, শূন্য আর এক — এই হল কম্পিউটারের তথ্য মানে ডেটা। এর মাঝে আমার মেশিনের হার্ডডিস্কটা ক্র্যাশ করল, দিন পাঁচ ছয় সাতে যে দুটো চল্লিশ জিবি করে হার্ডডিস্ক '/dev/hda' আর '/dev/hdb'-র কথা বলেছি, এর প্রথমটা গেল। একটা নতুন হার্ডডিস্ক কিনে আনলাম টাকা ধার করে, তবে দাম বেজায় কমে গেছে এখন, আশি জিবি ৭২০০ আরপিএম, তারই দাম নিল চার হাজার তিনশো। আরপিএম বিষয়টা মনে করতে পারছেন? সাত নম্বর দিনে ছিল। তা যাই হোক, নিজেই লাগালাম, বেজায় আমোদ পেয়েছি তাতে, নিজেকে বেশ কাবেল লাগছিল। দেখবেন, যে কাজটা আপনার এলাকার বাইরে, সেটা করতে পারলে ভারি ভালো লাগে। তা, এই একশোকুড়ি জিবি জোড়া মোট ডিস্কভূমির মোট ব্যবহৃত এই মুহূর্তে ৩২.৫ জিবি। এটা পেলাম সবগুলো পার্টিশনে মাউন্ট করার পর '`df -h`' মেরে, প্রতিটি পার্টিশনের ব্যবহৃত ভূমিগুলো যোগ করে। ৩২.৫ গুণ ১০২৪ এমবি, গুণ ১০২৪ কেবি, গুণ ১০২৪ বাইট, গুণ ৮ বিট। মোট দাঁড়াল, ১২০ জিবির ভিতর ৩২.৫ জিবি —

১.০৩০৮×১০^{১২} বিটের ভিতর ২.৭৯১৭×১০^{১১} বিট

মানে, মোটামুটি ধরুন ২৮-এর পরে দশটা শূন্য দিয়ে যত বড় সংখ্যা ততগুলো ছোট ছোট খোপ, যাতে হয় ১ আছে নয় ০ — এটা হল আমার মেশিনে এই মুহূর্তে বসবাসমান মোট তথ্যের পরিমাণ। আটাশ হাজার কোটি কুচো কুচো চৌম্বক খোপ যার প্রত্যেকটায় হয় ১ থাকতে পারে নয় ০। খোপের সংখ্যাটাকে বাড়িয়ে তোলা যাবে, আর হার্ডডিস্ক না বাড়িয়ে, এক লক্ষ কোটি অব্দি। একটু চ্যাংড়ামি করে বলা যায়, আমার মেশিনের মোট হার্ডডিস্কভূমিতে মোট যত শূন্য বা এক লেখা যায়, সেটা যদি আমায় হাতে করে লিখতে হত, আর মিনিটে যদি আমি একশোটা শূন্য বা এক লিখতে পারি, এই পোস্টচল্লিশে আর কত স্পিডে লিখব, আমার মাত্র ১৯০২৬ বছর লাগত। মানে এক বছরের মধ্যে গোটাটা লিখতে হলে, নিজে লেখার পর, আমার মাত্র উনিশ হাজার পঁচিশ জন সহকারী লাগত। তিন নম্বর দিনে আমরা লিখেছিলাম মনে আছে, অ্যাবাকাসের উদ্ভবের কথায়, নিজে এইসব করতে করতে মানুষের মনে হল, একটু সহকারী রাখার। সেই একটু সহকারী মানে মাত্তর উনিশ হাজার পঁচিশ জন। তাও এটা শুধু লেখার জন্যে। ব্যারন দ প্রনির হিউম্যান কম্পিউটারদের মোট কতজনের হিশেবের কাজ করে এক একটা মেশিন, সেই হিশেবটা নয় কিন্তু এটা।

কাজের কথায় আসা যাক। এই আটাশ হাজার কোটি খোপ জুড়ে কী আছে? আছে তথ্য। দুরকম তথ্য। এক, যে তথ্যকে চটকাতে হবে, ধরুন তাকে বলছি কাঁচা বা র তথ্য। আর, আগের সেকশনে যেমন বললাম, কী ভাবে এদের চটকাতে হবে, সেই আদেশ মালা, তারাও তো তথ্য, কম্পিউটারের নাড়িভুঁড়িতে তারাও তো ভরা আছে এই শূন্য আর এক দিয়েই। কারণ কম্পিউটারটা বেজায় গাধা, এই শূন্য বা একের, ভগবান আর শয়তানের, বাম আর রামের বাইনারি ছাড়া আর কিছু ব্যাটা বোঝেই না, মধ্যের ধূসরকে বুঝবে কী করে, আমাদের মত পোস্টমডার্নিজম পড়েনি তো। এই নির্দেশ তথ্য, ইন্সট্রাকশন ডেটা, সেটাও তো তথ্য, যাকে নয়মান স্টোরড প্রোগ্রাম বলে ডেকেছিলেন। এই প্রোগ্রাম মানে আমরা জানি, নানা জাতের নানা কিসিমের বাইনারি। কেউ শুধু সাঙ্গীতিক তথ্য চটকান মানে অডিও প্যাকেজ, কেউ শুধু লিখিত শব্দ চটকান মানে ওয়ার্ড প্রসেসর, কেউ শুধু মেশিন থেকে মেশিনে ভ্রাম্যমান তথ্য চটকান মানে ব্রাউজার, ইত্যাদি।

তাহলে কম্পিউটারের মোট ইতিহাস এবং ভূগোল কী দাঁড়াল? তার পেটের মধ্যে ভরা কোটি কোটি অর্বুদ অর্বুদ শূন্য বা এক লেখার খোপ — এটা তার ভাণ্ডার। এবং তার কাজ মানে কী? এই কোটি কোটি খোপে ভরা আরব আরব শূন্য বা এক ইধার-কা-মাল-উধার করা। ধরুন গাণিতিক একটা প্রতিতুলনা ব্যবহার করা যাক। এই অগণ্য শূন্য বা এক, এরা থিতু হয়ে আছে, এর মানে তথ্য। তথ্য তখন স্ক্যালার। আর বৌবাজার জেলেপাড়ার বাজারের শানে ‘কাঁটাপোনা লরচে’, মালেরা নড়ছে — এক ঠাঁই থেকে আর এক ঠাঁই যাচ্ছে, এর মানে এদের চটকানো হচ্ছে, মানে আমাদের মেশিন চলছে, কাজ করছে, প্রোগ্রামরা রান করছে। তথ্য তখন ভেক্টর। তার একটা গতি আছে দিকনির্দেশ আছে। তুলনাটা খুব একটা জমল কি? বুঝতে পারছিনা, সঙ্কর্ষণদের কাছ থেকে গালাগাল খাওয়া বা না-খাওয়াতেই বুঝতে পারব। আর কোনোদিন আমি এদের মত ছোটলোকদের সঙ্গে মিলে কিছু লিখছি না। আজ ভোর সোয়া পাঁচটায় আমায় ফোন করে বলেছে, এই সব খাজা মাল ছড়াচ্ছ কেন? দেড় দশকের বড় একটা লোকের সঙ্গে কথা বলার কী সহবত।

এবার দিন নম্বর একের সেই ওল্টানো সৌধের মত হার্ডওয়ার থেকে সফটওয়ারের হায়েরার্কির ছকটা আর একবার তুলে দিই এখানে। মনে করুন, একদম নিচের স্তরে, হার্ডওয়ারের স্তরে ভৌত চৌম্বকতার শূন্য আর এক, তাদের সঙ্গে অন্য ভৌত উপাদান, এর উপরে মাইক্রোআর্কিটেকচারের স্তরের ভিতটা, যেখানে দাঁড়িয়ে আছে মেশিন ল্যাংগুয়েজ, এখানেই হার্ডওয়ার বা যন্ত্রপাতির এলাকা শেষ। এবার মূল কাঠামো বা

(৬) হিসাবনিকাশের সফটওয়ার	(৬) রেলের টিকিট বিলিব্যবস্থা	(৬) ইন্টারনেটে ঘোরাঘুরি	প্রায়োগিক সফটওয়ারগুলো বা অ্যাপ্লিকেশনস
(৫) কম্পাইলার	(৫) এডিটর	(৫) কমান্ড ইন্টারপ্রিটার	মূল কাঠামো বা সিস্টেম প্রোগ্রামের এলাকা
(৪) অপারেটিং সিস্টেম			
(৩) মেশিন ল্যাংগুয়েজ			হার্ডওয়ার বা যন্ত্রপাতির এলাকা
(২) মাইক্রোআর্কিটেকচার			
(১) ভৌত উপাদানগুলো			

সিস্টেম প্রোগ্রামের এলাকা শুরু। এর মধ্যে আবার দুটো স্তর, নিচের ভিতটা হল অপারেটিং সিস্টেম, আমাদের গ্নু-লিনাক্স কারনেল। তার উপরে তিনটে সমান্তরাল জিনিষ — কম্পাইলার, এডিটর, কমান্ড ইন্টারপ্রিটার। আমরা এখন এদের চিনি, কম্পাইলার মানে আমাদের জিসিসি, এডিটর মানে ইম্যাক্স ভিম গোছের কেউ, আর কমান্ড ইন্টারপ্রিটার, হেইডা কেডা বলেন তো? এই দিনের আলোচনায় খাড়াইয়া আপনি যদি না-পারেন, তাইলে ব্যাস, খ্যামা দ্যান, মেশিনে গান শোনেন, তার বেশি কিছু আপনারে দিয়া হইব এমনডা লাগে না। আর এখানেই শেষ দ্বিতীয় স্তর। তার উপরের স্তরে আছে প্রায়োগিক সফটওয়ারগুলো, প্রতি মুহূর্তে যাদের নিয়ে আমরা কাজ করছি। নানা কাজের নানা বাইনারি বা বাইনারি-সমাহার।

তাহলে দেখুন, ভৌত চৌম্বকতার ওই শূন্য বা একের আকারেই তথ্যকে দেখাটা ফুরিয়ে যাচ্ছে হার্ডওয়ারের স্তরেই, তার পরে থাকছে, সরাসরি আমরা যেভাবে কাজ করি, কিবোর্ডে টাইপ করে যে বর্ণমালা পাই, তাদের সমাহার, এবং নিউলাইন, ক্যারেজ রিটার্ন, বেল ইত্যাদি আরো কিছু চিহ্ন। এদের মিলিয়ে যেটা তৈরি হচ্ছে সেটা আমাদের কাছে বোধ্য, হাইলেভেল হিউম্যান আন্ডারস্ট্যান্ডেবল ভাষায়। যেমন এডিটর কম্পাইলার এবং ব্যাশ ইত্যাদি কমান্ড ইন্টারপ্রিটারে আমরা যেভাবে কাজ করি। আমাদের পড়ার এবং বোঝার মত ভাষায় বলেই আমরা তাদের নিয়ে কাজ করতে পারি। শূন্য নম্বর দিনের তথ্যের পরিমাপ এবং অ্যাসকি সংক্রান্ত আলোচনাটা মনে করুন। কিন্তু এই কম্পাইলার, এডিটর বা ব্যাশ তথা যে কোনো কমান্ড ইন্টারপ্রিটারের মূল বাইনারি বা এদের উপরের স্তরে চালনীয় প্রোগ্রামগুলোর বাইনারি খুলে দেখুন, সেটা আমাদের সম্পূর্ণ অবোধ্য। যেমন, শূন্য নম্বর দিনেই আমরা একটু ভিশুয়াল চ্যাংড়ামি করেছিলাম, ছবি মবি দিয়ে ভিত্তার করে দেওয়া আমাদের ওই বাহারি পিডিএফ, সেটাকে সরাসরি এডিটর দিয়ে খুলে আমরা পাচ্ছিলাম ক্রিপ্টন গ্রহের সংখ্যাগুরু জনগোষ্ঠীর মাতৃভাষায় লেখা কাব্যগ্রন্থ, অথচ সেই জিনিষই পাঠ করে আমাদের সামনে পিডিএফ খুলে দিচ্ছে এক্সপিডিএফ বা গোস্টভিউ ইত্যাদি। কিন্তু আমাদের কাছে বোধ্য হোক, বা অবোধ্য, এই সমস্ত তথ্যই হল চিহ্ন দিয়ে তৈরি, চিহ্ন সমাহার। এই চিহ্ন সমাহার একটা প্রবাহের মত। যখন সেটা স্থির হয়ে আছে তখন সেটা তথ্য, কাঁচা তথ্য বা নির্দেশ তথ্য, র ডেটা বা প্রোগ্রাম। আর যখন মেশিন চলছে, প্রোগ্রামগুলো রান করছে, তখন সেই চিহ্নপ্রবাহ নড়ছে, একটা নির্দিষ্ট গতিমুখে নির্দিষ্ট নিয়ম মেনে, যে নিয়মগুলো ওই প্রোগ্রামে লেখা আছে।

তার মানে কী দাঁড়াল? কম্পিউটারে শেষ অব্দি কী আছে? চিহ্নের প্রবাহ। ক্যারেকটার স্ট্রিম। আর তাই যদি হয়, তাহলে, ভন নয়মানের ছকে আমাদের ‘cc’ মানে সচেতন নিয়ন্ত্রণের ব্যাটন হাতে নেওয়া মাত্র, ব্যাশকে স্থির করে নিতে হবে, তার চিহ্ন প্রবাহকে সঠিকভাবে প্রবাহিত করার মূল খাতগুলো কী হবে। মূল চলাচলপথের দিকগুলো প্রথম থেকেই ঠিক করে না-দিলে, পরে, প্রোগ্রামগুলো চলাকালীন, চারিদিকে প্রলয়নৃত্য পরায়ন নটরাজের মুদ্রার মত বহুমুখী বিচিত্রমুখী নিরবচ্ছিন্ন চিহ্নপ্রবাহের ট্র্যাফিক সে আর সামলাতে পারেনা। ঠিক এটাই ব্যাশ করে নেয় তার তিনটে প্রাথমিক ফাইল স্ট্যান্ডার্ড ইন, স্ট্যান্ডার্ড আউট আর স্ট্যান্ডার্ড এরর দিয়ে।

স্ট্যান্ডার্ড ইনপুট, স্ট্যান্ডার্ড আউটপুট এবং স্ট্যান্ডার্ড এরর। তথ্যের দিকনিয়ন্ত্রণের মূল উপাদান এই তিনটে। শুধু তথ্যের নয়, আপনার এবং আপনার দাসানুদাস ব্যাশেরও ক্রিয়ার দিকনির্দেশের তিনটে গুরুত্বপূর্ণ ভেক্টর এই স্ট্যান্ডার্ড ইন, সচরাচর ছোট আকারে ‘Std. In’, স্ট্যান্ডার্ড আউট, ‘Std. Out’, এবং স্ট্যান্ডার্ড এরর, ‘Std. Err.’। সেই পাঁচ নম্বর দিন থেকে আমরা যে বারবার উল্লেখ করছি, আলোচনা করছি, এমনকি না-জেনেই ব্যবহারও করছি, রিডাইরেকশন বা চালান করা এবং খাত বদলানো বা পাইপ করার কথা, সেটা এবারে আমরা ঠিক করে বুঝতে পারব।

আগেই অনেকবার বলেছি, গ্নু-লিনাক্সে সবই এক একটা ফাইল। এই স্ট্যান্ডার্ড ইন আউট এবং এরর — এরাও তিন পিস ফাইল। তিনটে ক্যারেকটার স্ট্রিম — চিহ্নের প্রবাহ। শূন্য নম্বর দিনেই বলেছিলাম, ক্যারেকটার মানে শুধু অক্ষর না, কমা স্পেস ইত্যাদি যতিচিহ্নগুলো, এবং নিউলাইন ক্যারেজ রিটার্ন বেল ইত্যাদি অছাপনীয় চিহ্নগুলোও। তার মানে, আরো ভেঙে বললে বাইটের স্রোত, তিনটের জন্যে তিন রকম স্রোতের তিনটে ফাইল। যে ফাইলগুলোকে ব্যাশ তথা ব্যাশাধীন বিভিন্ন কমান্ড নিজের ইনপুট বা আউটপুট হিশেবে ব্যবহার করে।

সঠিক টেকনিকাল অর্থে বলতে গেলে, কোনো কমান্ডই কিন্তু আপনার কিবোর্ডের কাছ থেকে আদেশ নেয়না, নেয় স্ট্যান্ডার্ড ইন থেকে। ডিফল্ট সেটিং-এ এই স্ট্যান্ডার্ড ইন হল আপনার কিবোর্ড। বা, ওই একই টেকনিকাল অর্থে, কোনো আদেশ বা কমান্ড তার নিজের কাজের ফলাফল আপনার স্ক্রিনে পাঠায় না, পাঠায় স্ট্যান্ডার্ড আউটে, যা

ডিফল্ট সেটিং-এ আপনার স্ক্রিন। আর, কমান্ড বা আদেশ পালন করতে গিয়ে যা যা ত্রুটি ঘটে, তাদেরকে পাবলিকলি জানান দেয় ব্যাশ তার প্রমাদ-বার্তা বা এরর মেসেজ দিয়ে, যার নাম স্ট্যান্ডার্ড এরর। এখুনি এদের ব্যবহারের ব্যাকরণে ঢুকব আমরা, তার আগে এদের কাজকর্মের সঙ্গে একটু পরিচিততর হয়ে নিই, আসুন। আমরা নয় নম্বর দিনের এবং আজকের আলোচনায় বারবার 'cat' করে '.bashrc' ফাইলকে অন্য একটা একটা ফাইলে নিয়েছি। তার জন্যে কমান্ড দিতে হয় '`cat ~/.bashrc>bashrc.text`'। বা '`.profile`' ফাইলকেও করা যায় এরকম। আমরা করেছি বারবার। এবার তার বদলে কমান্ড দেওয়া যাক

```
cat - ~/.bashrc - ~/.profile>local.conf.text
```

এবার এন্টার মারুন। দেখুন, ঠিক '`cat`' কমান্ডটা দেওয়ার পরে যেরকম একটা প্রম্পটহীন কালো স্ক্রিন আসে, সেরকম এসেছে। এই যে '–' চিহ্নটা রাখা হয়েছে দুবার, এটা আসলে দুটো হেডিং বানানোর জন্যে। 'local.conf.text' ফাইলটায় আমরা দুটো ফাইল রাখছি, তাদের দুটো হেডিং দেব এবার, যাতে একটা থেকে অন্যটাকে আলাদা করা যায়। '–' চিহ্নটা ঠিক এটাই করবে। আপনি কালো স্ক্রিনে টাইপ করে যা যা দেবেন, সেই তথ্য এবং ফাইলগুলো পরপর রাখতে রাখতে যাবে 'local.conf.text' ফাইলে। প্রথমে টাইপ করে দিন '`file: '~/.bashrc'`', তারপরে একটা আস্ত লাইন পরপর '=' চিহ্ন টাইপ করে দিন, এবার পরপর দুবার এন্টার মারুন, যাতে একটা লাইন ফাঁকা রাখে '.bashrc' ফাইলের আগে, এবার '<Ctrl><D>' মারুন। দেখুন, স্বাভাবিক 'cat' করার সময় এতেই যেমন কমান্ড প্রম্পট ফেরত আসে, মানে, ক্যাট করার কাজ শেষ হল, এখানে তা হয়নি, এখনো সেই কালো স্ক্রিন। এবার আসলে দ্বিতীয় '–' চিহ্নটার আদেশ পালন করছে ব্যাশ। এবার একই রকম একটা হেডিং টাইপ করে দিন '.profile' ফাইলের জন্যে। এবার '<Ctrl><D>' মারলেই দেখবেন কমান্ড প্রম্পট ফেরত এলো। মানে গোটা কাজটা ফিনিশ। এবার বোঝার চেষ্টা করা যাক। প্রথমে 'local.conf.text' ফাইলে ব্যাশ ক্যাট করল '–' চিহ্ন অনুযায়ী স্ট্যান্ডার্ড ইনপুটকে, যার ডিফল্ট হল কিবোর্ড, মানে কিবোর্ড থেকে আপনি যা যা টাইপ করলেন। তারপরে একটা ফাইল, আবার স্ট্যান্ডার্ড ইনপুট, আবার একটা ফাইল। অর্থাৎ, যা আমরা বলছিলাম, কোনো কমান্ড তার ইনপুট নানা জায়গা থেকে নিতে পারে, স্ট্যান্ডার্ড ইনপুট বা ডিস্ক ফাইল। এমন কি, নিতে পারে অন্য কমান্ডের স্ট্যান্ডার্ড আউটপুট থেকেও।

একটা কমান্ডকে অন্য কমান্ডের আউটপুট থেকে নিজের ইনপুট নিতে আমরা দেখেছি। ছয় নম্বর দিনের ১ নম্বর সেকশনে আমরা যখন '`man -k CD | less`' করে 'CD' শব্দটা কোন কোন ম্যানুয়াল পেজে আছে সেই তালিকাটা পড়ছিলাম, বা '`man -k CD | grep 'audio'`' করে যখন সেই তালিকাটার কোন কোন লাইনে 'audio' শব্দটা আছে সেটা খুঁজে বার করছিলাম, তখন কী ঘটছিল সেটা বোঝার চেষ্টা করুন। 'less' নামের পেজারটা একটা কমান্ড, পাতার এককে একটা চিহ্নপ্রবাহ বা ক্যারেকটার স্ট্রিমকে দেখানোর। সেই '`less`' কমান্ডটা আমরা এখানে তার ইনপুট নিচ্ছে স্ট্যান্ডার্ড আউটপুট থেকে, মানে স্ক্রিন থেকে। স্ক্রিনে যা রেকারিং ডেসিমালের মত ধ্যার‍্যার‍্যার‍্যার‍্যা করে বেরিয়ে যেত, তাকে ভদ্রলোকের পাতে দেওয়ার মত করে বাটিতে বাটিতে সার্ভ করছে 'less'। আর, পরেরটায় আমরা ব্যবহার করছি 'grep', খোঁজার কমান্ড, খুঁজে খুঁজে সেই লাইনগুলোকে বার করে 'grep', যেখানে উদ্দীষ্ট শব্দটা আছে, এখানে সেটা 'audio'। কারণ আমরা শুধু অডিয়ো সিডির ব্যাপারস্যাপারই খুঁজছিলাম। এখানেও দেখুন, স্ক্রিন থেকে মানে স্ট্যান্ডার্ড আউট থেকে তার ইনপুট নিচ্ছে 'grep'। এবং 'less' বা 'grep' দুজনেই তাদের নিজের নিজের ইনপুটের উপর নিজের কাজকর্ম করার পর তাকে পৌঁছে দিচ্ছে কোথায়? স্ক্রিনে মানে স্ট্যান্ডার্ড আউটের ডিফল্টে। আবার এরা এদের নিজেদের আউটপুটকে স্ট্যান্ডার্ড আউটে না-দিয়ে একটা ডিস্ক ফাইলেও দিতে পারত, ওখানেই সেটা করেছিলাম আমরা। পরে ধীরেসুস্থে পড়ার জন্যে একটা টেক্সটফাইল বানিয়ে নিয়েছিলাম।

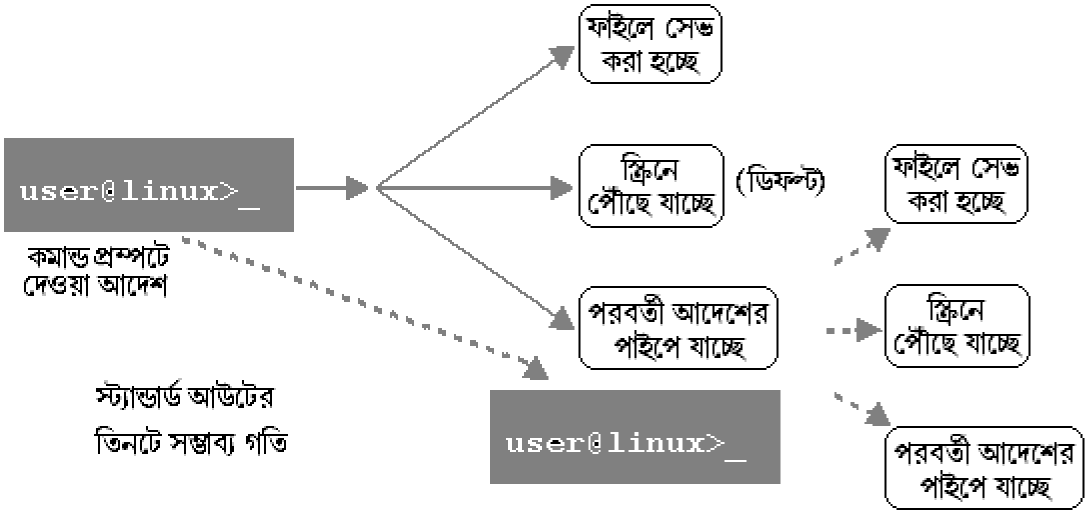
```
man -k CD | grep 'audio'> audio.cd.program.text
```

অর্থাৎ, একটা কমান্ডের কাছে কিছু এসে যায়না, তার ইনপুট কোথা থেকে আসছে — স্ট্যান্ডার্ড ইন না কোনো ডিস্ক ফাইল। বা কোথায় তার আউটপুট পৌঁছে দেবে — স্ট্যান্ডার্ড আউট না কোনো ডিস্ক ফাইল। স্ট্যান্ডার্ড ইন, স্ট্যান্ডার্ড আউট, স্ট্যান্ডার্ড এরর — এরা প্রত্যেকেই এক একটা ফাইল, ক্যারেকটার স্ট্রিম বা চিহ্নের প্রবাহ। স্ট্যান্ডার্ড ইন, স্ট্যান্ডার্ড আউট আর স্ট্যান্ডার্ড এরর — এদের তিনজনের ডিফল্ট তো বলেছি আগেই, প্রথমটার উৎস কিবোর্ড, আর

পরের দুটোর মোক্ষ বা চরিতার্থতা হল কনসোল বা স্ক্রিন বা টার্মিনাল। পরের দুটো যে একই ক্ষুরে মাথা মোড়াবে সেটা এভাবেই ব্যবস্থা করা। এই মাত্র আমরা দেখলাম, এবং এখুনি আরো ভালো করে বুঝব, যখনই ব্যাশ কোনো একটা বিশেষ চিহ্ন পায় বা আদেশ পায়, তখন, কী ভাবে কমান্ড বা আদেশগুলোর ডিফল্ট উৎস বা মোক্ষ বদলে ফেলে। এর কিছু চিহ্ন বা কায়দা আমরা এর মধ্যেই জেনেছি। যেমন '>' বা '<' বা '|' ইত্যাদি। এই বদলে ফেলাটাই হল রিডাইরেকশন বা চালান করা। আর রিডাইরেকশনটা যখন ঘটে একটা কমান্ড থেকে আর একটা কমান্ডের কাছে, তখন আমরা বলি প্রথম কমান্ডের আউটপুট বা ফলাফলকে দ্বিতীয় কমান্ডের কাছে পাইপ করা হল বা পথ-দেখানো হল। এই 'চালান' এবং 'পথ-দেখানো' প্রতিশব্দদুটো আমরা এনেছিলাম পাঁচ নম্বর দিনের ৪ নম্বর সেকশনে। খুব একটা চলেবল তা নয়, ঠিক আছে, মানে-টা তো বোঝা যাচ্ছে। আমাদের লাগের অরিজিত অঙ্কুর বাংলার সূত্রে একটা বড় প্রতিশব্দ তালিকা বানানোর প্রস্তাব করেছে, দেখা যাক কী হয়। অঙ্কুর বাংলার প্রতিশব্দগুলো আমারও ভাল লাগেনি, কিন্তু ছেলেগুলোর কোনো দোষ নেই, যে পরিমাণ খাটছে, আর ওরা কম্পিউটারটা যেমন জানে, ভাষা বা ব্যাকরণ ততটা ভালো জানার কথাও নয় ওদের।

১২।। চিহ্নপ্রবাহের গতিপথের চালচিত্র

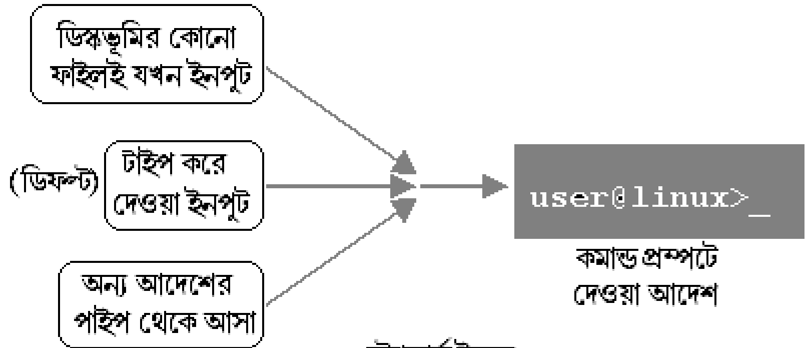
একটা আদেশ থেকে তৈরি হয় কিছু ফলাফল, কিছু চিহ্ন প্রবাহ, ক্যারেকটার স্ট্রিম, যা একটা ফাইল, তার নাম স্ট্যান্ডার্ড আউটপুট। এই স্ট্যান্ডার্ড আউট নামের চিহ্নপ্রবাহের ফাইলটার ভবিষ্যত হতে পারে তিন রকম। এক, তার ডিফল্ট গতি, অগতির গতি হল স্ক্রিন মানে কনসোল মানে মনিটর। আপনাকে চোখে গরম দেখিয়ে দিল তার কাজের ফলাফল। দুই, ফলাফল বা আউটপুটটা সরাসরি চলে গেল একটা ডিস্কফাইলে। আপনার ডিস্কভূমিতে নির্দিষ্ট কোনো একটা ফাইলে সঞ্চিত হল, সেভ করা হল। তৃতীয় গতিটা হল ত্রিশঙ্কু, মানে সরাসরি কোনো আউটপুট তৈরি হলনা, একটা কমান্ডের আউটপুট পাইপ হয়ে গেল আর একটা কমান্ডের ইনপুট হয়ে — আর একটা আদেশ যাকে আপনি দিয়েছেন ওই কমান্ড প্রম্পট থেকেই, সেই কমান্ড এবার প্রথম কমান্ডের আউটপুটটাকে নিজের মত করে চটকাবে, মানে প্রথম কমান্ডের থেকে তৈরি চিহ্নপ্রবাহের স্ট্যান্ডার্ড আউট এবার দ্বিতীয় তথা পরবর্তী কমান্ডের ইনপুট হয়ে



গেল। এবার সেই কমান্ড চটকানোর পরে আবার এই তিনটে গতি। এক স্ক্রিন, দুই ফাইল, তিন পুনঃপাইপিত হওয়া। এই গোটা প্রক্রিয়াটাকে দেখুন, এই ছবিতে দেখানোর চেষ্টা করেছি।

এখানে দেখুন, প্রথম কমান্ডের প্রক্রিয়াটাকে দেখিয়েছি স্বাভাবিক দাগে, পরের আদেশ থেকে সঞ্জাত প্রক্রিয়াটাকে দেখিয়েছি সচ্ছিদ্র দাগে, ডটেড লাইনে। আর দুটো কমান্ড প্রম্পটের মধ্যে যোগটা দেখিয়েছি ওই রকম দাগে, তার কারণ, কমান্ড প্রম্পট তো আপনি দুবার ব্যবহার করছেন না, ব্যবহার করছেন একবারই, যেমন একটু আগে আমাদের দেওয়া `man -k CD | grep 'audio'> audio.cd.program.text` কমান্ডটা ভাবুন। 'man' আর 'grep' আদেশদুটো তো দেওয়া হয়েছে একই সঙ্গে। এবার দেখুন 'man' আদেশটা তার ফলাফল পাঠিয়ে দিয়েছে পরবর্তী কমান্ড মানে 'grep'-এর কাছে, মানে, আমাদের ছবিতে একদম নিচের দাগটা। এটা হল পাইপ করা, বা পথ-দেখানো, যার জন্যে আমরা চিহ্ন দিয়েছি '|'। এবার 'grep' আবার তার ফলাফল থেকে তৈরি স্ট্যান্ডার্ড আউটকে

এবার পাঠিয়েছে 'audio.cd.program.text' নামে ফাইলের কাছে, মানে ছবিতে একদম উপরের সম্ভাবনাটায়। এটা হল রিডাইরেক্ট বা চালান করা, যার চিহ্ন দিয়েছি আমরা '>'। ব্যাশক্রিয়া এখানেই শেষ। ব্যাশকে আমরা অশেষ করার চেষ্টায় আরো আরো কমান্ড লাগাতে পারতাম। ধরুন `'man -k CD|grep 'audio'|less'`। মানে ফাইলে না-পাঠিয়ে এবার স্ট্যান্ডার্ড আউটের গতি হল তৃতীয় একটা কমান্ডের কাছে, 'less'। তার কাছে সেটা ইনপুট, 'less' এবার নিজের মত করে সেই চিহ্নপ্রবাহের পিণ্ড চটকে পৌঁছে দিল স্ক্রিনে, আমরা লেস দিয়ে পাতা বাই পাতা তাকে দেখতে পেলাম। তার মানে এবার, এতক্ষণে, দুটো কমান্ডের পাইপের পথ পেরিয়ে স্ট্যান্ডার্ড আউট তার ডিফল্ট গতিপ্রাপ্তিতে পৌঁছলো।



স্ট্যান্ডার্ড ইনের
তিনটে সম্ভাব্য উৎস

স্ট্যান্ডার্ড আউটপুটের টার্মিনাসের সংখ্যা তিন, আর স্ট্যান্ডার্ড ইনপুটের উৎসের সংখ্যা তিন। আগের স্ট্যান্ডার্ড আউটের ছবিতে দেখুন, দিকনির্দেশটা হল কমান্ড থেকে স্ট্যান্ডার্ড আউটের দিকে চিহ্নের প্রবাহ, আর এবারের স্ট্যান্ডার্ড ইনের ছবিতে দেখুন, এখানে চিহ্নের প্রবাহটা স্ট্যান্ডার্ড ইন থেকে কমান্ডে। কমান্ড প্রম্পটে টাইপ করে দেওয়া আপনার কোনো আদেশ তার ইনপুট হিশেবে ব্যবহার করছে যে চিহ্নপ্রবাহ বা ক্যারেকটার স্ট্রিমকে, তার অন্য নাম স্ট্যান্ডার্ড ইন। আদেশের কাছে এই স্ট্যান্ডার্ড ইন আসার ডিফল্ট উপায়টা হল টাইপ করে দেওয়া ইনপুট। এছাড়া আসতে পারে আপনার ডিস্কভূমিতে ইতিমধ্যেই সেভ করে রাখা কোনো ফাইল থেকে। আসতে পারে অন্য কোনো আদেশের পাইপ মারফতও, যেখানে সেই পূর্ববর্তী সমাধিত আদেশের কাজের ফলাফল বা আউটপুট হল এই চালু বর্তমান আদেশের ইনপুট। যেমন আগের সেকশনেই ব্যবহৃত `'man -k CD|grep 'audio'|less'` কমান্ডটা ভাবুন। এখানে 'less' আদেশের কাছে ইনপুট আসছে 'grep' কমান্ডের ফলাফল, আবার 'grep'-এর কাছে এসেছিল 'man' কমান্ড থেকে। এই ভাবেই একটা আদেশের পাইপ বেয়ে অন্য আদেশের কাছে যাচ্ছে চিহ্নপ্রবাহ। কিন্তু যদি এই কাজটা আমরা দুটো ভাগে ভেঙে নিতাম?

ধরুন, যদি, কমান্ড দিতাম, `'man -k CD > cdtemp; grep 'audio' cdtemp|less'`, তাহলে কী হত? এখানে দেখুন প্রথমে আমরা 'man' কমান্ডের ফলাফলটাকে চালান করে দিচ্ছি একটা ডিস্কফাইলে। আচ্ছা, বলুন তো, হঠাৎ, আগের সেকশন থেকে, ফাইলকে ডিস্কফাইল বলে ডাকতে শুরু করলাম কেন? কারণটা যদি নিজেই ভেবে নিতে পারেন, বোঝা যাবে আপনার গ্নু-লিনাক্সের অভ্যেশ ভালোই চলছে। গ্নু-লিনাক্সে সবই ফাইল, তাই, ক্যারেকটার স্ট্রিম বা চিহ্নপ্রবাহ — সেটাও একটা ফাইল। কিন্তু সেই ফাইল আমাদের প্রাত্যহিক চেনা ফাইল নয়, তার আবাস সাইবার প্রক্রিয়ার ভৌতিক ভূমিতে — ভারচুয়াল করিয়া তারে করেছো এ কী সন্ন্যাসী, বাফারময় দিয়াছ তারে ছড়ায়ে? যখন আমরা ওই চিহ্নপ্রবাহকে রিডাইরেক্ট করি আমাদের ডিস্কভূমিতে ন্যস্ত ডিরেক্টরিগুলোর কোনো একটার ভিতর কোনো একটা নির্দিষ্ট নামে, নির্দিষ্ট কিছু বাইট লিখে ফেলা হয় ওই চিহ্নপ্রবাহের অন্তর্বস্তু দিয়ে, তখন সেই ফাইলটা আমাদের প্রাত্যহিক ব্যবহৃতব্য ফাইল হয়ে ওঠে। ডিস্কফাইল নাম দিয়ে এটাই বোঝাচ্ছি যে এটা ওই ধরনের কোনো সাইবারবাসী ঘনচক্কর ফাইল নয়, নিতান্তই ভাতডাল খাওয়া ডিস্কবাসী ফাইল, এমনকি পকেট হাতড়ালে এক আধটা বনগাঁ লাইনের মাস্থলি টাস্থলিও বেরোতে পারে।

হ্যাঁ, যা বলছিলাম, আগের কমান্ডটা দিয়ে 'man' কমান্ডের ফলাফলটাকে স্ট্যান্ডার্ড আউটের ডিফল্ট স্ক্রিনে না নিয়ে চালান করে দিচ্ছি বর্তমান কাজের চালু ডিরেক্টরিতে, মানে যে ডিরেক্টরিতে বসে কমান্ডটা দিচ্ছি সেখানে, 'cdtemp'

নামের ডিস্কফাইলে। এর পরে দেখুন একটা ‘;’ চিহ্ন, ঠিক যে যতিচিহ্ন হিশেবে আমরা সেমিকোলন ‘;’ ব্যবহার করি, সেই কাজই করছে এখানে। শেষ এই কমান্ডটা আসলে পরপর দুটো কমান্ডের একটা সমাহার। পরপর কাজ করার দুটো কমান্ডকে আলাদা করে এই ‘;’ চিহ্ন। এবার পরের কমান্ডটুকুকে আলাদা করে বোঝার চেষ্টা করুন।

```
grep 'audio' cdtemp | less
```

একটু আগে ‘grep’ কমান্ডটা তার ইনপুট পাচ্ছিল সরাসরি ‘man’ আদেশের ফলাফল থেকে। এখন আর তা নয়। ‘grep’ আদেশটা এখন তার ইনপুট নিচ্ছে ‘cdtemp’ নামের ডিস্কফাইল থেকে। সেই ফাইলের মধ্যে ‘audio’ শব্দটা কোন কোন লাইনে আছে সেটা খুঁজে দেখছে। তারপর সেই লাইনগুলোকে তুলে আনছে সামনে মানে আউটপুটে। সেই আউটপুটকে আমরা আবার ‘less’ দিয়ে পড়ছি। তার মানে, ‘grep’ কমান্ডকে ইনপুট হিশেবে যে উৎসই আমরা দিই না কেন — তা সে ‘man’ কমান্ডের ফলাফলের পাইপ হোক, বা সেই ফলাফল দিয়ে গজিয়ে তোলা ডিস্কফাইল ‘cdtemp’ হোক, তাতে ‘grep’-এর কাজ করার কোনো পার্থক্য ঘটেনা।

স্ট্যান্ডার্ড ইনের ছবিটায় দেখুন, তিনটে উৎসের মধ্যে দুটো, উপরের আর নিচেরটা আমরা চিনলাম, মধ্যেরটা এখনো বাকি, মানে, ইনপুটটা যখন আসছে কিবোর্ড থেকে, মানে স্ট্যান্ডার্ড ইনের যেটা ডিফল্ট। কমান্ড প্রম্পটে টাইপ করে দেওয়া চিহ্নপ্রবাহ যখন কমান্ডের ইনপুটের উৎস। কিন্তু সেটাও আসলে বাকি নেই। আগেই দেখানো হয়ে গেছে। চিহ্নপ্রবাহের এই ফাইল তিনটেকে নিয়ে আলোচনা শুরু করার সময়, আজকের ১১ নম্বর সেকশনে। সেখানে আমরা কমান্ড দিয়েছিলাম, ‘`cat - ~/.bashrc - ~/.profile>local.conf.text`’। এই কমান্ডটা ঠিক কী কাজ করছে, কেমন ভাবে, সেই আলোচনা ওখানেই আছে। ‘cat’ কমান্ডটা একবার তার ইনপুট একবার পাচ্ছে ‘-’ চিহ্ন দিয়ে দেখানো কিবোর্ড ইনপুট থেকে, একবার পাচ্ছে ‘~/.bashrc’ ইত্যাদি ডিস্কফাইল থেকে। কিবোর্ড বা ডিস্কফাইল যে উৎস থেকেই আসুক, তাতে ব্যাশের কিছু এসে যাচ্ছেনা।

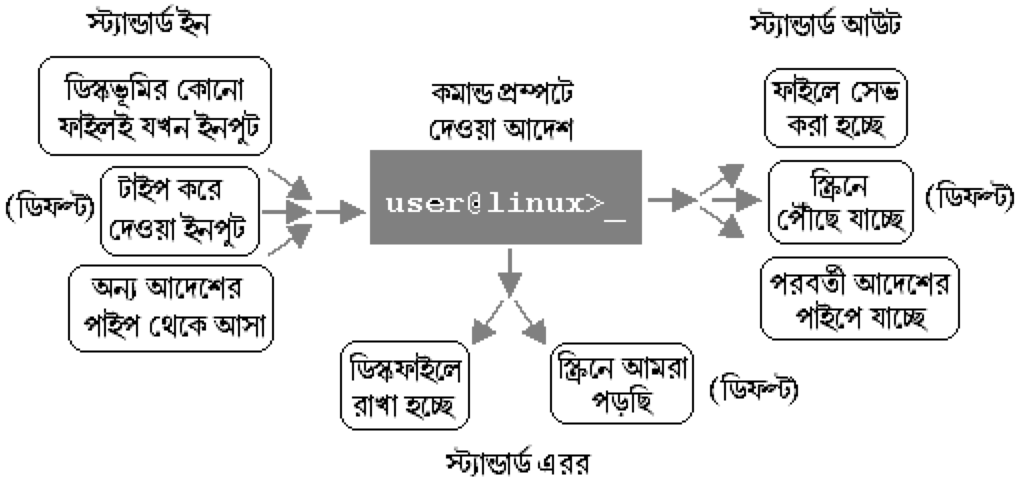
এই স্ট্যান্ডার্ড ইন আর আউটেই কিন্তু ব্যাশচরিত শেষ হয়না, আদেশ পালন করতে গিয়ে বহু রকম প্রমাদ ঘটে, মানে, আদেশ পালনে সিস্টেমের তথা ব্যাশের অপারগতা। খুব সহজটাই ভাবুন, এমন একটা ফাইল আপনি কনক্যাটেনেট করতে, ‘cat’ করতে, বললেন, যে নামের কোনো ফাইলই ওই ডিরেক্টরিতে নেই, যে ডিরেক্টরিতে দাঁড়িয়ে আপনি কমান্ডটা দিয়েছেন। ধরুন, এই আগের সেকশনেই আমরা যে ‘cdtemp’ ফাইলটা বানিয়েছিলাম, সেটা স্বাভাবিকভাবেই, নামে যেমন বোঝা যাচ্ছে, একটা অস্থায়ী বা টেম্পোরারি ফাইল, তাই কাজ মিটে যাওয়ার পরেই আমরা উড়িয়ে দিয়েছি। এবার, এখন যদি হঠাৎ করে কমান্ড দিই, ‘`cat cdtemp`’, স্বাভাবিকভাবেই, ব্যাশ সেই নামের কোনো ফাইল আপনার হোম ডিরেক্টরিতে খুঁজে পাবেনা, ফাইলটাই তো আর নেই। তাহলে ব্যাশ এখন কী করবে? সে আপনাকে প্রমাদ বার্তা বা এরর মেসেজ পাঠাবে। ব্যাশ আপনার কনসোলে ফুটিয়ে তুলবে,

```
cat: cdtemp: No such file or directory
```

এক একটা আদেশ পালনে সিস্টেমের এক এক ধরনের অপারগতার জন্যে এক এক রকম প্রমাদ বার্তা। এই প্রমাদ বার্তাটা আপনি একটা ফাইলেও পাঠাতে পারেন, তারও উপায় আছে, সে কথায় আসছি আমরা। আপাতত এই কমান্ডটা দেখুন, ‘`cat cdtemp 2>errorfile`’। এর আগে আমরা যখন শুধু ক্যাট করার কমান্ডটা দিচ্ছিলাম, আমাদের যে প্রমাদবার্তাটা ব্যাশ স্ক্রিনে দেখাচ্ছিল সেই বার্তাটাকেই এখন চালান করে দিচ্ছে ‘errorfile’ নামের ডিস্কফাইলে। চালান করার ওই চিহ্নটা, ‘>’, আমরা চিনি, শুধু এখানে ওই ‘2’ সংখ্যাটা আমরা চিনিনা, দুনম্বরিটা চিনে যাব, ক্রমে আমরা ব্যাশিক্ষিত হচ্ছিনা? এবার যদি আমরা ওই ডিস্কফাইলটাকে স্ক্রিনে পড়ি, ‘`cat errorfile`’ কমান্ড দিয়ে, ঠিক সেই এরর মেসেজটাই পাব, যেটা এর আগে ব্যাশ আমাদের সরাসরি স্ক্রিনে দেখাচ্ছিল।

ব্যাশের এই প্রমাদ বার্তাই হল চিহ্নপ্রবাহের তিন নম্বর ফাইল, স্ট্যান্ডার্ড এরর। ছবিতে দেখুন, উপরের বাঁদিকের স্ট্যান্ডার্ড ইন, বা ডানদিকের স্ট্যান্ডার্ড আউট আমাদের চেনা, এর সঙ্গে এসেছে স্ট্যান্ডার্ড এরর, যার আবার দুটো গতি, হয় স্ক্রিনের ডিফল্ট, নয়ত রিডাইরেক্ট করা ডিস্কফাইল। এইমাত্র যেটা আমরা করে দেখলাম। ব্যাশ আমাদের প্রমাদবার্তা পাঠিয়েছিল, কারণ আমরা ‘`cat cdtemp`’ কমান্ড দিয়ে ‘cdtemp’ নামের এমন একটা ফাইল ‘cat’ করতে বলেছিলাম, যে ফাইলটাই নেই। তার থেকে তৈরি হয়েছিল প্রমাদ বা এরর। যে বার্তা বা মেসেজটা আমরা

পরে ‘errorfile’ নামের ফাইলে পাঠিয়েছিলাম, পুরোনো ‘`cat cdtemp`’ কমান্ডটার সঙ্গে ‘`2>errorfile`’ বলে একটা বাড়তি অংশ যোগ করে।



এই ‘2’ সংখ্যাটাকে বলে ফাইল ডেসক্রিপ্টর। ফাইল ডেসক্রিপ্টর দিয়ে আমরা ইন, আউট আর এরর — এই তিন স্ট্যান্ডার্ড ক্যারেকটার স্ট্রিম ফাইলকে চিনে থাকি। তিনটের জন্যে তিনটে আলাদা আলাদা সংখ্যা। ‘0’ মানে স্ট্যান্ডার্ড ইন, ‘1’ মানে স্ট্যান্ডার্ড আউট আর ‘2’ মানে স্ট্যান্ডার্ড এরর। চালান বা রিডাইরেক্ট করার চিহ্ন ‘>’-এর সঙ্গে এই তিনটে সংখ্যার যে কোনোটাকে লাগানো মানে সেই চিহ্নপ্রবাহটাকে রিডাইরেক্ট করা। যেখানে তার ডিফল্টে যাওয়ার কথা, সেখানে যেতে না-দিয়ে আমরা অন্য কোথাও নিয়ে যাচ্ছি তাকে, চালান করছি, রিডাইরেক্ট করছি। এইমাত্র আমরা যে ‘`2>errorfile`’ অংশটা যোগ করলাম, তার মানে, এরর বার্তাটাকে তার ডিফল্ট গতি মানে স্ক্রিনে যেতে না-দিয়ে আমরা ঠেলে দিচ্ছি একটা ফাইলে, যার নাম ‘errorfile’। তার আগে অব্দি প্রমাদবার্তাটা আমরা স্ক্রিনে পাচ্ছিলাম। এখানে স্ট্যান্ডার্ড এরর বোঝাতে ‘2’ ফাইল ডেসক্রিপ্টরটা ব্যবহার করতে হল, কিন্তু, ‘0’ বা ‘1’-এর বেলায় সচরাচর করতে হয়না, কারণ সেটাই ডিফল্ট। ধরুন আমরা চালু ডিরেক্টরির সমস্ত ফাইলের তালিকা বানাচ্ছি ‘ls’ কমান্ডের স্ট্যান্ডার্ড আউটকে রিডাইরেক্ট করে, ‘`ls > filelist`’ কমান্ড দিয়ে। এখানে আমরা যা আদতে করছি, তা হল, ‘`ls 1> filelist`’। কিন্তু ‘1’-টা আলাদা করে উল্লেখ করতে হয়নি, কারণ ওটাই ডিফল্ট। কখনো কখনো সেটা করতে হয় বৈকি, একাধিক প্রবাহকে একই সঙ্গে একটা ফাইলে রিডাইরেক্ট করতে হলে। কিন্তু সে ব্যাশের অনেকটা অগ্রবর্তী কমান্ডের ব্যবহারে। খুব যদি কৌতূহল হয়, তো, খেজুরগাছে হাঁড়ি বাঁধো ম্যান, ম্যান পড়ার নেশা করো ম্যান।

১৩।। কমান্ডের মধ্যে কমান্ড

একটা কমান্ডের তৈরি আউটপুট যখন সরাসরি আর একটা কমান্ডের ইনপুট হয়ে আসে, সেটা আমাদের চেনা। আমরা সেই পাঁচ নম্বর দিন থেকে আজ অব্দি অজস্রবার ব্যবহার করে আসছি। একে আমরা ডাকি পাইপ করা বা পথ-দেখানো। যার চিহ্ন ‘|’। বারবার করেছি ব্যবহার। এর আলোচনায় আর যাচ্ছিনা। এই পাইপ ছাড়া আরো কয়েকভাবে একটা কমান্ডের মধ্যে আর একটা কমান্ড ব্যবহার করা যায়। আজকের আলোচনার ৯ নম্বর সেকশনের একদম শেষে এসে আমরা এর একটার কথা উল্লেখও করেছিলাম, ব্যাককোট বা ‘``’। এই চিহ্নটা পাবেন কিবোর্ডের একদম বাঁয়ে উপরে, ঠিক ‘<Esc>’ বা ‘এসকেপ’ চাবিটার নিচেই। আমরা এই চিহ্নটার আলোচনা এনেছিলাম ‘writefiles’ নামে ব্যাশস্ক্রিপ্টের সূত্রে। লাইনটা ছিল ‘``c=`expr $c + 1` ``’। এই স্ক্রিপ্টে আমরা একটা কাউন্টার ভ্যারিয়েবল ব্যবহার করেছিলাম, যার নাম ‘c’, যার আলোচনা আমরা নয় নম্বর দিনেই করেছি, ২.২ নম্বর সেকশনে, স্ক্রিপ্টের লাইন ধরে ধরে আলোচনায়, একবার দেখে নিন। এই লাইনটার মোদ্দা কাজ হল ‘c’ নামের ভ্যারিয়েবলটার মান প্রতিবার ঘোরার সঙ্গে সঙ্গে এক করে বাড়ানো। ঘোরা বলতে ‘writefiles’ স্ক্রিপ্টে দেখুন, একটা একটা করে ফাইল ক্যাট করে তুলে আসা। এর আগের লাইনটাই দেখুন, ‘`cat $i>> $f`’, যেখানে ও একটা ফাইলকে তুলে দিচ্ছে। আর তার পরেই ‘c’-এর মান এক বাড়িয়ে দিচ্ছে। কেন বাড়াচ্ছে? কারণ, তার আগের লাইনে যেখানে প্রতিটি ফাইল

ক্যাট করার আগে একটা করে হেডিং বানাচ্ছে, সেখানে ফাইলের নম্বরটা লিখে দিচ্ছে ব্যাশ। স্ক্রিপ্টটার গোড়াতেই দেখুন অ্যাসাইন করা আছে, ‘c=1’। তার মানে প্রথম ফাইলের হেডিং-এ এটাই আসবে, ফাইলটা ক্যাট করার পর ‘c’-এর মান এক বাড়িয়ে দেবে, তার মানে পরের ফাইলে মানটা আসবে দুই, এই ভাবে বাড়তে থাকবে। এই কাউন্টার ব্যাপারটা প্রোগ্রাম লেখার একদম গোড়ার একটা খুব প্রাথমিক তরকিব।

এবার লাইনটাকে বোঝার চেষ্টা করা যাক — কী ঘটছে লাইনটায়। ‘expr’ হল একটা অঙ্ক কষার কমান্ড। সিস্টেমেই থাকে। আপনার প্রিয় ম্যানুয়ালদের এক নম্বর সেকশনেই দেখুন দেওয়া আছে এর ইতিবৃত্ত। প্রথমে দুই ব্যাককোট চিহ্নের মধ্যের অংশটাকে ভাবুন। ‘expr $c + 1’। এটা নিজেই আলাদা একটা কমান্ড। অ্যাসাইন করার কমান্ড। আমরা আগেই বলেছি ‘=’ চিহ্নটা অ্যাসাইন বা ধার্য করে। এখানে গোটা প্রক্রিয়াটা ভাবুন। প্রথমে, স্ক্রিপ্টের একদম গোড়াতেই আপনি ব্যাশকে জানিয়েছেন, ‘c=1’, অর্থাৎ ‘c’-এর মান ধার্য করো ‘1’। অর্থাৎ এইমুহূর্তে ব্যাশ জানে ‘c’ হল ‘1’, মেশিনের স্মৃতিপটে ‘c’ নামক ভ্যারিয়েবলের জন্যে বরাদ্দ করা জমিতে সে লিখে রেখেছে ‘1’ সংখ্যাটা। ভ্যারিয়েবলের এই অ্যাসাইন করার ব্যাপারটা তো আমাদের পরিচিত। এবার সে প্রথম ফাইল ক্যাট করে দিল। করার পরে এল আমাদের উদ্দীষ্ট লাইনে। এখানে সে পাচ্ছে ‘expr $c + 1’। ‘expr’ কমান্ডটা নিছক অঙ্ক কষে। তাকে আমরা বললাম, তুমি ‘$c’ মানে ‘c’ ভ্যারিয়েবলের মানের সঙ্গে ‘1’ যোগ করো। যোগ করে সে কী পাবে? তার কাছে আছে, ‘c’-এর মান হল ‘1’, এর সঙ্গে সে আরো ‘1’ যোগ করবে, তার মানে, যোগফল হবে ‘2’।

ধরুন ওই স্ক্রিপ্টের ভিতরে নয়, বাইরে, একদম ব্যাশ কমান্ড প্রম্পটে এই গোটা কাজটা আমরা করছি। যদি পরপর এই দুটো কমান্ড দিই আমরা, প্রথম লাইনে একটা ভ্যারিয়েবল অ্যাসাইন করি, ‘c=1’, আর তার পরের লাইনে দিই, ‘expr $c + 1’, তাহলে ব্যাশ স্ক্রিনে ফুটিয়ে তুলবে, ‘2’। মানে অঙ্কটা সে কষে ফেলল, ‘expr’ কমান্ডটা দিয়ে যে অঙ্কটা তাকে কষতে বললাম আমরা। কিন্তু আমাদের কাজ এখনো শেষ হয়নি। এই মুহূর্তে আমরা ব্যাশ প্রম্পটে যদি দিই, ‘echo $c’, ব্যাশ আমাদের কী দেখাবে? ব্যাশ দেখাবে ‘1’, কারণ ‘c’ ভ্যারিয়েবলের মান ওটাই, ব্যাশের কাছে তাই দেওয়া আছে। এবার, আমাদের উদ্দীষ্ট ওই গোটা লাইনটা ব্যাশের কমান্ড প্রম্পটে আদেশ আকারে দেওয়া যাক, ‘c=`expr $c + 1`’। এবার আর একবার ‘echo $c’ দিয়ে দেখুন। এবার দেখবেন, ব্যাশ ফুটিয়ে তুলল, ‘2’। অর্থাৎ, ব্যাককোট বা ‘``’ চিহ্নের মধ্যের যে অংশটা, যার মান ‘2’ আমরা এইমাত্র দেখেছি সেটাকে অ্যাসাইন করে দেওয়া হয়েছে ‘c’ ভ্যারিয়েবলের জন্যে বরাদ্দ করা জমিতে। তাহলে, ‘=’ চিহ্ন দিয়ে ভ্যারিয়েবল অ্যাসাইনমেন্ট কমান্ডের মধ্যে আমরা আর একটা কমান্ড ‘expr’-কে ব্যবহার করে নিতে পারলাম এই ব্যাককোট চিহ্নের দৌলতে।

আমাদের স্ক্রিপ্টটাকে এবার ভাবুন, প্রতিবার, ক্যাট করার পরের লাইনেই, ব্যাশ কী করছে? ‘c’ ভ্যারিয়েবলের সেই মুহূর্তের মান পড়ে নিচ্ছে মেমরি থেকে, সেই মানটার সঙ্গে ‘expr’ কমান্ড ব্যবহার করে ‘1’ যোগ করে নিচ্ছে, তারপরে এই যোগফলটাকে অ্যাসাইন করে দিচ্ছে ‘c’ ভ্যারিয়েবলের জন্যে বরাদ্দ করা জমিতে। মানে প্রতিবার ‘c’ ভ্যারিয়েবলটা এক করে বেড়ে যাচ্ছে। তাই প্রত্যেকটা ফাইলের হেডিং-এ আসা সংখ্যাটা আগের ফাইলের সংখ্যাটার থেকে এক বেশি হচ্ছে। আমরা ঠিক যেভাবে গুনি, সেই গোনার প্রক্রিয়াটা আমাদের ব্যাশ স্ক্রিপ্টিং ভাষা দিয়ে আমরা ব্যাশের হাতে দিয়ে দিচ্ছি। নয়মান সাহেবের ‘CC’ বা কেন্দ্রীয় নিয়ন্ত্রণ এখন মেশিনের ভিতরে চলে গেছে, ক্যালকুলেটরে যা ছিলনা।

আদেশ বদল বা কমান্ড সাবস্টিটিউশনের এই কাজটা আমরা আর এক ভাবেও করতে পারতাম। আমাদের ইতিমধ্যেই চেনা ‘$’ চিহ্ন দিয়ে। আমাদের ‘writefiles’ স্ক্রিপ্টে যদি ‘c=`expr $c + 1`’ লাইনটা ইম্যাক্স দিয়ে মুছে, তার জায়গায় এই লাইনটা যোগ করে দিই, ‘c=$(expr $c + 1)’, স্ক্রিপ্টটা একই ভাবে কাজ করবে। নিজের মুন্ডুর ভিতরকার প্রক্রিয়াগুলোর ‘CC’ যদি আপনি অন্য কারো হাতে বকলমা না-দিয়ে থাকেন, তাহলে, এখানে কী ঘটছে সেটা আপনি নিজেই বুঝে নিতে পারবেন। শুধু মাথায় রাখবেন, ‘$’ মানে মান। প্রথমে করছি ‘$c’ অর্থাৎ ‘c’-এর মান, তার সঙ্গে ‘expr’ দিয়ে ‘1’ যোগ করছি। এবার এই গোটাটাকে দেখছি একটা ভ্যারিয়েবল হিশেবে, যার মান বার করছি এই গোটাটা দিয়ে — ‘$(expr $c + 1)’। এবং সেই মানটাকে অ্যাসাইন করে দিচ্ছি ‘c’ ভ্যারিয়েবলের জন্যে বরাদ্দ করা জমিতে তার পুরোনো মান মুছে দিয়ে।

কমান্ডের মধ্যে কমান্ড টেনে আনার শেষ প্রক্রিয়াটা বেড়ে মজার। কমান্ডটার নাম ‘tee’, এতে গাছেরটা খাওয়া যায়, এবং একই সঙ্গে, একটুও না-থমকিয়ে বা না-চমকিয়ে তলারটাও কুড়োনো যায়। আমাদের আজকের আলোচনার ২

নম্বর সেকশনে '.bashrc' ফাইল থেকে কয়েকটা লাইন তুলে দিয়েছিলাম আমরা, সেখানে বলেছিলাম শেষ চারটে লাইন এই মুহূর্তে খ্যামা দিন। সেই লাইন চারটের গোটাটা বোঝার ক্যালি আমাদের এখনো হয়নি। কিন্তু মধ্যের একটা লাইন এবার পেড়ে ফেলা যাক।

```
echo -e '\n***\n'>>F;/usr/bin/fortune -s | tee -a F | cat
```

এই লাইনটায় ব্যবহার হয়েছে মোট চারটে কমান্ড, 'echo', 'fortune', 'tee', এবং 'cat'। 'echo' কমান্ডের পরে '-e' অপশনটার মানে আমরা জানি, যাতে '\n' দেখলে একো সত্যিই একটা করে লাইন বাদ দেয়। একোকে আদেশ দেওয়া হয়েছে, তুমি 'F' নামের ফাইলে প্রথমে যোগ করো একটা ফাঁকা লাইন, তার পরের লাইনে তিনটে তারকাচিহ্ন বা অ্যাসটেরিক্স '***' যোগ করো, তার পরে ফের একটা ফাঁকা লাইন যোগ করো। খেয়াল করুন, এখানে আমরা রিডাইরেকশনের '>' ব্যবহার না-করে, অ্যাপেন্ড বা যোগ করার '>>' চিহ্ন দিয়েছি। এর মানে আমরা জানি, শুধু '>' চিহ্নে ডিরেক্টরিতে থাকা পুরোনো 'F' ফাইল মুছে নতুন একটা 'F' নামের ফাইল খুলে নেবে ব্যাশ, কিন্তু '>>' থাকায় সেটা না-করে, পুরোনো ফাইল থাকলে তাতে যোগ করে দেবে, আর না-থাকলে ওই নামের একটা ফাইল বানিয়ে নেবে। এর পরে আছে একটা ';' চিহ্ন। তার পরে পরের কমান্ড '/usr/bin/fortune -s'। কমান্ড প্রম্পটে দিলে একটা করে বাণী শোনায় ফরচুন, এর কথা তো আগে বারংবার বলেছি। শুধু '-s' অপশনটা ফরচুনকে জানায়, একটু ছোট করে কাকা, শর্টে মেরো। আর '/usr/bin' অংশটা হল 'fortune' নামের বাইনারির ঠিকানা। তারপরে একটা পাইপ করার চিহ্ন '|'। তার মানে ফরচুন বাণীটাকে আমি সরাসরি স্ক্রিনে না-দেখে পাইপ করে দিচ্ছি। এটা খুব চেনা প্রক্রিয়া। ধরুন এরকম যদি কমান্ড দিই আমরা, 'fortune -s >> F' — এতে ফরচুন বাণীটা স্ক্রিনে না-দেখে আমরা 'F' ফাইলে চালান করে দিই। হয় স্ক্রিন নয় ফাইল। দুটো এক সঙ্গে হয় না। কিন্তু এই পাইপ চিহ্নের পরে 'tee' কমান্ডটা এবার দিচ্ছে সেই মজা, কেকও খাবো, আবার তার ডোটাও রয়ে যাবে। 'tee -a F' কমান্ডটা ব্যাশকে বলে দিচ্ছে, প্রথমে তুমি 'fortune' কমান্ডের কাছ থেকে পাইপ করে যেটাকে পেয়েছ সেটাকে 'F' ফাইলে যোগ করো, ওই '-a' অপশনটার মানে হল অ্যাপেন্ড বা যোগ করা। কিন্তু মজাটা দেখুন, এটা 'tee' না-হয়ে 'cat' হলে কমান্ডটা এখানেই শেষ হয়ে যেত, নতুন করে পাইপ করার আর কিছু থাকত না, স্ট্যান্ডার্ড আউটটা পৌঁছে যেত 'F' ফাইলে। কিন্তু এখানে তা হলনা, 'tee' যে চিহ্নপ্রবাহটুকু যোগ করল 'F' ফাইলে, সেটাকে আবার রেখেও দিল। পাইপ করে দিল কমান্ডের একদম শেষ অংশে, 'cat'। তার মানে, এবার আমরা তাকে স্ক্রিনে পড়তে পেলাম। একই সঙ্গে 'F' নামের ফাইলে ফরচুন বাণীগুলো যোগ হতে থাকল, এবং সেগুলো লগ-ইন করা মাত্র আমি স্ক্রিনে পড়তেও পেলাম। আপনাদের কারোর যদি দরকার পড়ে, এভাবে জমানো অফেনসিভ ফরচুন বাণী আমার ফাইল করে রাখা আছে, সেটা আপনাদের দিতে পারি, মেল করবেন।

এইবার আমরা ঢুকব ব্যাশ নিয়ে আমাদের আলোচনার একদম শেষ অংশে। ব্যাশের কয়েকটা অবস্থানগত নিয়ন্তা বা পোজিশনাল প্যারামিটার নিয়ে কথা বলব, কয়েকটা অতিচিহ্ন বা মেটাক্যারেকটার, একটু লুপ ছোঁব, তার পরেই আমাদের খেল খতম, জিএলটি ইশকুল পাঠমালা হজম। আমি ছুটি পাব নভেম্বর এক থেকে আজ আঠেরোই মার্চ অব্দি আমার এই ছোটার হাত থেকে। হার্ডডিসকের কারণে কয়েকটা দিন ছেড়ে দিলে আর কলেজের কাজগুলো ছেড়ে দিলে আমি উনিশ ঘন্টার বেশি নষ্ট করিনি এই কটা মাসে।

১৪।। অবস্থানগত নিয়ন্তা — এবং তাদের বদলানো

এর আগে ছয় নম্বর দিনে এবং আজকেরও গোড়ার দিকে আমরা ব্যাশের আভ্যন্তরীন ডিফল্ট রকমে সিস্টেম ভ্যারিয়েবলগুলোকে চেনার কাজে 'set' বলে একটা কমান্ড ব্যবহার করেছি। এটা 'set' কমান্ডের দিক থেকে দেখলে, অত্যন্ত অপমানজনক। ধরুন একটা তিমির একটা ডানাই যদি তিমিটার পরিচিতি হয়ে দাঁড়ায়। কিম্বা, আপনি বললেন, মাছি নিয়ন্ত্রণে লিটলবয় এবং ফ্যাটম্যানের কোনো তুলনা হয়না, হিরোশিমা নাগাসাকিতে অন্তত বছরখানেক কোনো মাছি ছিলনা। এর আগেও আমরা ব্যাশের আভ্যন্তরীন বিল্ট-ইন কাজগুলোর কথা এনেছি। ওরফ বা অ্যালিয়াস, কিম্বা, রপ্তানি বা এক্সপোর্ট ইত্যাদির আলোচনার সূত্রে। ব্যাশের ম্যানুয়ালেই পেয়ে যাবেন 'set' নামের বিল্ট-ইন কমান্ডের এবং অবস্থানগত নিয়ন্তা বা পোজিশনাল প্যারামিটারগুলোর কথা। সেট আদেশটাকে আপনি একা একলা কমান্ড প্রম্পটে দিয়ে এন্টার মারলে কী হয়, সেটা দেখেছি আমরা। সমস্ত চালু ব্যাশ ভ্যারিয়েবলের, ফাংশনের একটা তালিকা

তৈরি করে দেয় সামনে। কিন্তু এই সেট দিয়ে একটা বিপুল কাজ করা যায়, নিজের প্রয়োজনমাফিক কিছু নিয়ন্তা বা প্যারামিটারকে ব্যাশের কাছে পৌঁছে দেওয়া, এর পরে আপনার কাজে বারবার যে নিয়ন্তাগুলোকে নিজের খুশি মত ব্যবহার করতে পারবেন।
প্রথমে একটা অর্থহীন উদাহরণ দিয়ে ব্যাপারটা বুঝে নেওয়া যাক। একটা কমান্ড দিন

```
set Arab kafela is a procession of man camel struggle and broken dreams
```

এন্টার মারুন। দেখুন, কমান্ড প্রম্পট ফেরত এসে গেছে, তার মানে যা কিছু ঘটার ঘটে গেছে। এটা ব্যাশেষত্ব। মানে ব্যাশের বিশেষত্ব। ও একজন আদ্যোপান্ত নীরব কর্মী। যে কাজটা আপনি করতে দিয়েছেন, সেটা নীরবে করে দেবে। রব তৈরি করবে একমাত্র তখনই যখন কোথাও কোনো একটা প্রমাদ ঘটবে। স্ট্যান্ডার্ড এররকে তার ডিফল্টে মানে স্ক্রিনে শো করে দেবে তখন। এখানে ব্যাশ কোনো উচ্চবাচ্য করল না, মানে, যা করার করে দিয়েছে। কিছু একটা ঘটে গেছে মেশিনের ইলেকট্রনিক পেটে। এখুনি যা মালুম পাওয়া যাবে। এবার কমান্ড দিন

```
echo `whoami` $3 $4 $8
```

হুবহু যা লেখা তা যদি টাইপ করে দিতে পারেন, আপনার আইডেন্টিটি ক্রাইসিসের একদম হাতে হাতে একটা সমাধান পেয়ে যাবেন। পেলেন? খুব বেশি খচে যাওয়ার কোনো কারণ নেই, এমন নয় যে ফাঁকতালে আমি আপনাকে গাল দিয়ে নিলাম। ভাবুন, এটা এখানে লেখার আগে করে দেখে নিতে গিয়ে হুবহু ওই আত্মজ্ঞান আমারও পেতে হয়েছিল। আর এত চটার কী আছে? 'রক্তবরণ-মুগ্ধকরণ-নদীপাশে-যাহা-বিঁধিলে-মরণ' — তিনিই তো বলেছিলেন, 'উট উটঅ উঠঅ: সে তো মিস্টিরিয়াস অ্যানিমাল মশাই' (কার্টসি জটায়ু)।
এবার দেখা যাক, এখানে কী ঘটছে। আচ্ছা, 'whoami' কমান্ডটার কথা আগে কোথাও বলেছি? মনে পড়ছে না, আর খোঁজা তো অসম্ভব। এই ইতিমধ্যেই এক লাখ চৌষট্টির জগদ্দলে। না, এই 'whoami' কোনো গভীর দার্শনিক আত্মজিজ্ঞাসা নয়, নিতান্তই একটা চালু ব্যাশের চালু ব্যবহারকারীর ইউজার নেম ফুটিয়ে তোলার কমান্ড। সেটাকে আমরা দিয়েছি দুটো ব্যাককোটের মধ্যে। মানে, কমান্ড সাবস্টিটিউশন, আদেশ-বদল। কমান্ডের মধ্যে কমান্ডের সেই চক্কর, আমরা 'whoami' কমান্ডের ফলাফলটাকে এখানে 'echo' কমান্ডের মধ্যে টেনে আনছি। এবার তার পরের অংশটা, মানে, '$3 $4 $8' — এটা আমাদের সেট কমান্ডের অবদান। তার আগেই যে লাইনটা আমরা সেট কমান্ডে দিয়েছিলাম, সেই লাইনটায়, 'set' শব্দটার পর থেকে গুনুন, এক নম্বর শব্দ হল 'Arab', দু নম্বর 'kafela'। এরকম করে তিন চার আর আট নম্বর শব্দ তিনটে কী কী সেটা দেখে নিন। এবার আপনার আত্মজ্ঞানের রহস্যভেদ হল? আসলে 'set' ঠিক এটাই করে। তার পরে যে শব্দগুলো আসছে সেগুলোকে স্পেসে স্পেসে আলাদা করে এক দুই তিন চার — এই ভাবে পরপর সংখ্যায় অবস্থানগত নিয়ন্তা বা পোজিশনাল প্যারামিটার হিশেবে তুলে রাখে। এবার যখনই দরকার পড়ে, আমরা তাদের '$1', '$2', '$3' ইত্যাদি নামে ডাকি, এবং তাদের বৈদ্যুতিন অন্ধকার থেকে ডাকিনীমন্ত্রে জাগিয়ে তোলে ব্যাশ।

এবার এর থেকে একটু কম অর্থহীন একটা কাজ করানো যাক ব্যাশকে দিয়ে। আর তাতে, আমাদের '~/.bashrc' ফাইল বদলানোটাও একটু অভ্যেশ হবে। এই তিনটে লাইন আপনার '~/.bashrc' ফাইলের শেষে যোগ করে দিন

```
set $(date)
echo Hello Dear $(whoami)
echo Today is $1-day, $2, $6, and the time is: $4
```

এবার একবার লগ-আউট লগ-ইন করুন, বা, জ্যান্ত রকমে মানে ইন্টারাক্টিভলি একটা ব্যাশ শেল চালু করুন, ব্যাশের ভাষায় ইনভোক। মানে জাস্ট কমান্ড প্রম্পটে '`bash`' লিখে এন্টার মারুন। এটা আমি করায় ব্যাশ আমায় দেখাল

```
Hello Dear dd
Today is a Fri-day, Mar, 2004, and the time is: 10:58:09
```

এর মধ্যে একটা জিনিষ তো গত সেকশনেই শিখেছি, ব্যাককোট বা '``' চিহ্নের জায়গায় '`$()`' দিয়ে আর এক রকমের আদেশ-বদল বা কমান্ড সাবস্টিটিউশন। আমরা এই একই ফল পেতাম যদি আগের তিনটে লাইন এরকম লিখতাম

```
set `date`
```

```
echo Hello Dear `whoami`
echo Today is $1-day, $2, $6, and the time is: $4
```

এর তিন নম্বর লাইনটাকে বোঝার চেষ্টা করার আগে একবার 'date' কমান্ডটা ব্যবহার করে দেখে নিন, কী ফলাফল তৈরি হয়। আমি কমান্ড লাইনে 'date' লিখে এন্টার মারায় ব্যাশ স্ক্রিনে দেখাল

```
Fri Mar 19 10:59:11 IST 2004
```

'date' কমান্ডটা ঠিক এটাই করে, দিন তারিখ সময় তুলে দেয়। এবং লগ-ইন করার মুহূর্তে ব্যাশ যা পড়েছিল তার থেকে আমি 'date' কমান্ডটা দেওয়ার মধ্যে দেখুন এক মিনিট দু সেকেন্ড সময় চলে গেছে। প্রত্যেকবারই লগ-ইন করার সময় ব্যাশ একবার করে '~/.bashrc' ফাইল থেকে এই তিন লাইন কমান্ড পড়বে, এবং সেই অনুযায়ী ফুটিয়ে তুলবে। যেমন আমরা আগেই বলেছি, করার কথা ব্যাশের। এবং ব্যাশ কী রকমে অবস্থানগত নিয়ন্তা বা পোজিশনাল প্যারামিটারগুলো পড়ল, সেটা খেয়াল করুন। ব্যাশ নিজে 'date' কমান্ডটা দিয়ে যা পেয়েছিল, সেটা হল 'Fri Mar 19 10:58:09 IST 2004'। এবার স্পেস বা ফাঁকা জায়গার যতিটা মাথায় রাখুন। তাহলে ব্যাশের কাছে এখন '$1' মানে 'Fri', '$2' মানে 'Mar', '$3' মানে '19', '$4' মানে '10:58:09', '$5' মানে 'IST', এবং '$6' মানে '2004'। এবার জাস্ট আমাদের দেওয়া আদেশের তৃতীয় লাইনের দিকে তাকান, এবং দেখুন, ব্যাশ আর কিছুই করেনি, শুধু, '$1', '$2', '$6' এবং '$4' নামের অবস্থানগত নিয়ন্তাদের জায়গায় তাদের নিজের নিজের মান ভরে দিয়েছে। এবং ভাবুন, এই ভাবে পাওয়া অবস্থানগত নিয়ন্তাদের কত রকম ভাবে ব্যবহার করা যায়। যান করতে শুরু করুন, কোনো বাধা নেই, শুধু আপনার কল্পনাশক্তি এবং ব্যাশ ম্যানুয়ালই এর লিমিট। এই 'set' কমান্ড দিয়ে পাওয়া অবস্থানগত নিয়ন্তাদের আয়ু কত? পরের বার 'set' মেরে নতুন নিয়ন্তা বানানোর আগে অব্দি। অথবা, আপনি লগ-আউট করা অব্দি। প্রায় এই একই রকম ভাবে এই ধরনের কাজে ব্যবহারের আর একটা বেড়ে জিনিষ আছে ব্যাশে। তার নাম বিশেষ ব্যাশ চল বা স্পেশাল ব্যাশ ভ্যারিয়েবল।

১৫।। বিশেষ ব্যাশ ভ্যারিয়েবল

এই বিশেষ ব্যাশ ভ্যারিয়েবল বলার আগে একটা মাপ চেয়ে নেওয়ার আছে। এই বইয়ে বারবার বলেছি আমার কম্পিউটার করার স্বশিক্ষিত মানে অশিক্ষিত রকমের কথা, কোনো ভালো জায়গায় ভালো করে কিছু শেখা হয়নি, সেটা লাগের ছেলেমেয়েদের দেখলে বুঝতে পারি। এই দৌড়ে তো হলনা, পরের চান্সে ট্রাই করা যাবে, তবে ততদিনে আর কোনো হিজল-বট-জারুল এই বাংলায় থাকবে কিনা কে জানে। ব্যাশ বা শেল আমার নিজের কাছে বেশ উত্তেজক লাগে। সেই উত্তেজনাটা আমার মধ্যে চারিয়ে গেছিল স্টিফেন প্রাটার 'অ্যাডভান্সড ইউনিক্স এ প্রোগ্রামারস গাইড' (Stephen Prata: 'Advanced Unix — A Programmer's Guide') বলে একটা বই থেকে। সত্যি বলছি, একটুও বাড়িয়ে বলছি না, কলেজ থেকে ফেরার সময় আমি নিজে নটা-পাঁচের বনগা লোকালের অন্যের-কান-চুলকে-দেওয়া ভিড়ে দাঁড়িয়েও পড়ে দেখেছি, বেশ পড়া যায়, এত ভালো বই। ট্যানেনবম-এর মডার্ন অপারেটিং সিস্টেমের মত ক্রাইম থ্রিলার না হলেও মোটামুটি সাইফি সিনেমা বলে চালানো যায়। আমার কম্পিউটার ভাবনার যেটুকু যতটুকু — তা প্রায় আগাগোড়াই এই দুটো বইয়ের কল্যাণে তৈরি। কিন্তু এখানেই একটা ব্যথা আছে। এ দুটো বইয়ের একটাও গ্নু-লিনাক্সের জন্যে কাস্টম-বিল্ট নয়। ট্যানেনবমে তবু গ্নু-লিনাক্সের কোনো কোনো আলোচনা আছে। আর প্রাটার বইটার ভারতীয় সংস্করণই ছিয়াশির, মানে, লিনাস তখনো তার কারনেলই নামায়নি নেটে, পাঁচ নম্বর দিনের গোড়ার ক্রোনলজিটা দেখুন। আর প্রাটার এই বইটা বর্ন শেল নিয়ে লেখা। বর্ন-এগেন শেল মানে ব্যাশ নয়। ব্যাশে বর্নের সমস্ত কলকব্জাই আছে, কিন্তু শুধু সেটুকুই নেই, আরো কিছু আছে। সেই আরো কিছুটা আমি কখনো কখনো ব্যাশ ম্যানুয়াল থেকে দেখেছি বটে, কিন্তু বিস্তর ফাঁক রয়ে গেছে। আর আপনারা নিজেরাই একটা করে বই লিখুন, বাংলায়, গ্নু-লিনাক্সের, সত্যি বলছি, এই বইটা লিখতে লিখতে গোটাটাই আমার মাথায় এত স্পষ্ট হয়ে উঠেছে, কোনো কিছু শেখার জন্যে একটা বই লেখার মত ভালো কোনো উপায় আর হয়না। আর এভাবেই তো বেঁচে থাকি আমরা। সব কিছুর পরেও। শূন্য নম্বর দিনের গোড়াতেই ছিল দুটো বাচ্চার কথা, দমদম স্টেশনে মেট্রোর কাউন্টারের পাশে বসে ভিক্ষে করত। কাল প্রায় সেই বয়সেরই দুটো বাচ্চাকে দেখলাম মধ্যমগ্রাম স্টেশনে। সাত আট বছরের দিদি দেড় দু বছরের ভাইকে কোলে নিয়ে দু-নম্বর প্ল্যাটফর্ম থেকে নামল লাইনে, তারপর লাইন পেরিয়ে এক নম্বরে আসছে। প্রতিটি বার পা ফেলার পর ওর হাঁটু বেঁকে যাচ্ছে, মুখ কুঁকড়ে যাচ্ছে, পাথরে এতটাই লাগছে, কিন্তু তাতে

দায়িত্বপালনে কোনো খামতি ঘটছে না। আমরা পোস্টমডার্নিজমে মানব জাতির ইতিহাসকে ব্যক্তিমানবের অতিক্রম করে যাওয়ার গল্প বলি, তার এর চেয়ে বড় উদাহরণ তো আর হয় না। চারদিকে যখন এ ওরটা কেড়ে খাওয়া, অন্যের খিদের থেকে চোখ ফিরিয়ে রাখাটাই নিয়তি ভবিতব্য এবং সামাজিক কানুন হয়ে দাঁড়াচ্ছে, তখনো তো প্যাঁচা জাগে, পাথরের পথে বাচ্চাতরকে কোলে বাচ্চা দাঁতমুখ খিঁচিয়ে পথ হাঁটে। তাহলে আমরা পারিনা কেন? নিজেদের মতো করে নিজেদের রেজিস্ট্যান্সের ব্যাকরণ আসুন নিজেরাই বানাই। অনেক কিছু দিয়ে তা ঘটে। এর কোনো বাঁধাধরা নিয়ম নেই। আসুন গ্নু-লিনাক্স দিয়েও আমরা খিদে জিইয়ে রাখার রাজনীতির বিরুদ্ধে এক এক পিস প্রতিরোধ বানাই। আমাদের তো রামও নেই, যে পিঠে রামের পাঁচ-আঙুলের সমাদরের দাগ নিয়ে গাছের ডালে ডালে লেজ বেঁকিয়ে লাফিয়ে বেড়াব, না-থাকুক, সেতুবন্ধের কাজে নিজের নিজের ব্যক্তিগত বাদামের খোলা তো থাকতে পারে।

যাকগে, যা বলছিলাম, ব্যাশের নিজেরই বরাদ্দ করা বিশেষ কিছু ভ্যারিয়েবল। 'PATH' বা 'PS1' ইত্যাদি এনভায়রনমেন্ট ভ্যারিয়েবলকে ধারণ করার ব্যাশপ্রক্রিয়ার কথা তো আমরা আগেই বলেছি। এর সঙ্গে ব্যক্তিগত লোকাল ভ্যারিয়েবল বানানো এবং ব্যবহারের কায়দা নিয়েও কথা বলেছি আমরা। এবার আমরা আগে থেকেই নির্দিষ্ট কিছু বিশেষ ব্যাশ ভ্যারিয়েবলকে দেখব। অবস্থানগত নিয়ন্তা বা পোজিশনাল প্যারামিটারদের পরেই এদের কারণটা এখুনি দেখতে পাবেন। অঙ্কের সঙ্গে যাদের পরিচয়টা একটু কম, তাদের জন্যে একটু ভ্যারিয়েবল আর প্যারামিটার ধারণাদুটো নাড়িয়ে নিই। ধরুন একটা বাচ্চার রেজাল্ট নিয়ে ভাবছেন আপনি, তাকে বারবার বলছেন, পড়ার সময় বাড়া, সিনেমা দেখিস না, আড্ডা একটু কম মার। ইত্যাদি। কিন্তু একবারো বলছেন না, তোর বাবার আয়টা একটু বাড়া, তোর প্রতি তোর বাবা-মার ভালোবাসাটা একটু বাড়া, তোর চারপাশের বায়ুমণ্ডলে দূষণটা একটু কমা যাতে তোর হাঁফানিটা একটু কম হয়। কিন্তু নিজেই ভেবে দেখুন, এর সবগুলোই তো সমান রকমের গুরুত্বপূর্ণ। এর কারণ, এই মুহূর্তে আপনি চেয়ে চেয়ে মরে গেলেও, বা তার চেয়েও সাংঘাতিক, বিপ্লবী হয়ে গেলেও, এর কিছু কিছু জিনিষ আপনি কিছুতেই বদলাতে পারবেন না। আপনার সীমাবদ্ধ অধিকারের ভূমিতে, বাচ্চাটার লেখাপড়ার ভালো করার জন্যে মাত্র কয়েকটা জিনিষকেই আপনি বদলাতে পারেন। এগুলোই ভ্যারিয়েবল। আর যারা অপরিবর্তনীয় থাকছে, স্থির রয়ে যাচ্ছে, তারাই প্যারামিটার। আবার সিচুয়েশন থেকে সিচুয়েশনে এই ভূমিকাগুলোও বদলায়। ধরুন, আপনি সোজিওলজি ফিল্ডে লোক ঠকান, যেমন আমার পকেটমারার ব্যক্তিগত এলাকাটা হল অর্থনীতি, এবার আপনি একটা স্টাডি করছেন, সেখানে ব্যক্তি বাচ্চাটা আর নেই, আছে পরপর ইনকাম গ্রুপ, আর তাদের ঘরের বাচ্চাদের রেজাল্ট। এখানে রেজাল্ট নামক ফাংশনের ভ্যারিয়েবল হয়ে দাঁড়াল বাচ্চার মা-বাবার আয়, বাচ্চাটার ব্যক্তিগত পছন্দ অপছন্দ তথা আর সমস্ত কিছু এখন প্যারামিটার বা নিয়ন্তা। অন্য আরো প্রচুর উপাদানের সঙ্গে মিলে একটা অনড় স্থির গড় ব্যাপার হয়ে আছে প্রতিটি ইনকাম গ্রুপেই। ইকনমিক্সে এই ধরনের জন্যে সাইডলাইনের বাইরে বসে থাকা উপাদানগুলোর জন্যে একটা কাল্টনাম আছে — সিটেরিস পেরিবাস — অল আদার থিংস রিমেইনিং কনস্টান্ট — আর সমস্ত কিছু স্থির আছে এই অবস্থায়। ভারি নিরাপদ। যা যা বিষয় আপনার জন্যে অস্বস্তিকর, তাদের ঠেলে দিতে পারবেন এই সিটেরিস পেরিবাসে। এই এলাকার প্রতিটি উপাদানই তখন প্যারামিটার। যেমন, এই বাচ্চাটার উদাহরণে রেজাল্ট নামক ফাংশনের ভ্যারিয়েবল হল বাচ্চাটার পড়ার ইচ্ছে, দেখা সিনেমার সংখ্যা, আড্ডা মারার মোট সময়ের পরিমাণ। বাবার আয়টা তখন ছিল প্যারামিটার।

মানে, একটা বিশেষ পরিস্থিতিতে আপাত অপরিবর্তনীয় মালটা হল প্যারামিটার, আর বদলাতে থাকা বস্তুটা হল ভ্যারিয়েবল। এবার আমরা যে তালিকাটা আনব, তাতে দেখুন, একটা করে বিশেষ চিহ্ন বা চিহ্ন-সমাহার, আর তার বিশেষ একটা অর্থ। এই অর্থটা অপরিবর্তনীয়। সেই অর্থে তাই এদের নিয়ন্তা বা প্যারামিটার বলাই ভালো। কিন্তু ভ্যারিয়েবল নামটা চলে যখন, তার কোনো একটা আলাদা অর্থ বা ইতিহাস আছে নিশ্চয়ই, যেটা আমি জানিনা। ব্যাশ ম্যানুয়ালে দেখেছি, এদের স্পেশাল প্যারামিটার বলেই ডাকছে, ভ্যারিয়েবল নয়, আমাদের সাধারণ বুদ্ধিও তাই বলছে, এমনও হতে পারে যে প্রাটা ভুল করে এদের স্পেশাল ভ্যারিয়েবল বলে ডেকেছেন।

$# — 'set' কমান্ড দিয়ে আমরা যখন অবস্থানগত নিয়ন্তাদের মানগুলো দিয়ে দিচ্ছি ব্যাশকে, তখন আমাদের সেই আদেশে পোজিশনাল প্যারামিটারের মোট সংখ্যা। যেমন, একটু আগে, উট বলে ডাকার আমাদের উদাহরণটার সময়, আমরা যদি ব্যাশ প্রম্পটে '`echo $#`' কমান্ড দিতাম, ব্যাশ আমাদের দেখাত, '`12`'। গুনে

দেখুন, কেন এই সংখ্যাটাই দেখাচ্ছে ব্যাশ। তবে খেয়াল রাখবেন, পরের বার 'set' কমান্ডটা দেওয়ার আগে অব্দি বা লগ-আউট করার আগে অব্দি এটা লাগু থাকবে।

$* বা $@ — এরা দুটোই দেখায়, 'set' কমান্ড দিয়ে মোট যত অবস্থানগত নিয়ন্তা বা পোজিশনাল প্যারামিটার ব্যাশকে দেওয়া হয়েছে, তার তালিকা। দুটোর মধ্যে একটাই তফাত আসে, যখন কোট চিহ্নের মধ্যে ব্যবহার করা হয় এদের। যেমন আমরা ব্যাশে লগ-ইন করছি যখন, '~/.bashrc' ফাইলে ওই '`set $(date)`' কমান্ডের কারণে '`echo $*`' বা '`echo $@`' করলেই, লগ-ইনের মুহূর্তে 'date' কমান্ড যা দেখিয়েছিল, পোজিশনাল প্যারামিটারের জমিতে তখন ভরে রাখা সেই গোটাটা ব্যাশ দেখিয়ে দেবে আমাদের। করে দেখুন। বা উটের উপাখ্যানে, আরব কাফেলার বিষয়ে গোটা ওই শব্দবন্ধটা। কিন্তু প্রতিবার 'set' কমান্ড দেওয়া মাত্রই আগেরটা মুছে নতুন মান ভরে দেওয়া হবে পোজিশনাল প্যারামিটারের জমিতে।

$! — '`echo $!`' কমান্ড দিলে ব্যাশ আপনাকে দেখায় শেষ যে প্রক্রিয়া বা প্রসেসটা ব্যাকগ্রাউন্ডে গেছে, তার পিআইডি (PID) বা প্রক্রিয়া-সূচক। ব্যাকগ্রাউন্ডের আলোচনা আজকেরই ৪ নম্বর সেকশনে আমরা করে এসেছি। ধরুন আপনি একটা ইন্টারাক্টিভ শেল প্রক্রিয়া চালু করলেন, কিন্তু তাকে রাখলেন ব্যাকগ্রাউন্ডে। '`bash &`' কমান্ড দিয়ে। এই অ্যাম্পারস্যান্ড বা '`&`' চিহ্নটা তো আমরা চিনি, একটা প্রক্রিয়াকে সক্রিয় সম্মুখভূমি থেকে আপাতনিষ্ক্রিয় পশ্চাৎভূমিতে ঠেলে দেয়। যেই আপনি এটা করলেন স্ক্রিনে আপনাকে ব্যাশ এই জাতীয় কিছু একটা দেখাবে — '`[1] 1301`'। এর অর্থটা মনে করতে পারছেন, প্রথম চৌকো ব্রাকেটের মধ্যে '`[1]`' বস্তুটা হল জব নাম্বার বা বকেয়া কাজের সূচক, আর পরের '`1301`' হল এর পিআইডি। এবার আপনি যদি '`echo $!`' কমান্ড দিয়ে এই '$!' প্যারামিটারটা জানতে চান, ব্যাশ আপনাকে ওই পিআইডিটা দেখাবে, মানে, '`1301`'।

$$ — '`echo $$`' আপনাকে দেখায় এই মুহূর্তে চালু ব্যাশ শেলের পিআইডি। একদম হাতে নাতে পরখ করে নিন। এই মাত্র আপনি একটা ব্যাশকে ব্যাকগ্রাউন্ডে পাঠিয়েছেন। একে ফোরগ্রাউন্ডে আনতে তো আমরা জানি। জাস্ট '`fg`' করুন। বা, আরো কাউকে কাউকে আপনি যদি পশ্চাৎপদ এলাকায় পানিশমেন্ট পোস্টিং করে থাকেন, তাহলে তার ব্যাকগ্রাউন্ড ব্যাশের জব নম্বর দিয়ে করুন। ব্যাশ চালু হয়ে গেল। এখন আপনি আছেন আমাদের সেই চেনা ঘাপলা ব্যাশের পেটে ব্যাশে। তাতে অবশ্য ফারাক কিছু হবেনা ব্যাশের আচার-আচরণে। এবার কমান্ড দিন '`echo $$`'। দেখুন ব্যাশ আপনাকে দেখিয়ে দিল, সেই '`1301`', আপনি তো চেনেন, ব্যাকগ্রাউন্ডে পাঠানোর সময় এইমাত্র ব্যাশ আপনাকে দেখিয়ে দিয়েছিল ব্যাশের পেটে ব্যাশের ওই নতুন প্রক্রিয়াটার পিআইডি।

$0 — এই মুহূর্তে ব্যাশ যে কমান্ডটা পালন করছে, তার নাম ভরা থাকে এই বিশেষ প্যারামিটারে। এটার একদম হাতেগরম একটা প্রয়োগ হতে পারে, আমাদের একটু আগের ৯ নম্বর সেকশনের 'writefiles' নামের ব্যাশস্ক্রিপ্টটাই। ধরুন, একটু একটু বদলে আপনি ওই একই রকম কাছাকাছি কাজের বেশ কয়েকটা আরো স্ক্রিপ্ট বানিয়েছেন, 'writefiles1', 'writefiles2' ইত্যাদি নামে। এবার, 'writefiles' স্ক্রিপ্টটার একদম শেষ লাইনে, মানে, '`done`' লাইনটায় লুপ শেষ করার পর, যদি আর একটা ঠিক এই রকম লাইন যোগ করেন, '`echo -e "\n\n$d\n[Made by $0 on $(date)]">>$f`', তাহলে কী হবে বলুন তো? বেশ একটা মেড-ইন-ইন্ডিয়া গোছের লাগছে না? এখানে আমি ইচ্ছে করে এটা বোঝালাম না, ওখানের আলোচনার সঙ্গে মিলিয়ে পড়লে নিজেই বুঝতে পারবেন। ফলাফলটা হবে এই যে, এইরকম একটা অংশ 'writefiles' দিয়ে বানানো ফাইলের শেষে যোগ হয়ে যাবে —

```
******************************************
[Made by ~/bin/writefile on Sat Mar 20 05:16:22 IST 2004]
```

$? — এটা শেষ ব্যাবহৃত ব্যাশ কমান্ডের এক্সিট স্ট্যাটাস বা নিষ্ক্রমণ স্থিতি দেখায়। মানে, কমান্ডটা তার কাজ করতে পেরেছে কি পারেনি। এদিকে ব্যাশ বেশ নাগার্জুনপন্থী, মানে তার কাছে সাফল্য হল শূন্য। যদি '`echo $?`' করলে '`0`' দেখায়, তার মানে কাজটা হয়েছে। সি-তেও কিন্তু তাই।

$- — ব্যাশ শেলের অপশান। '`echo $-`' কমান্ডটা দেখায়, যে ব্যাশের ভিতরে দাঁড়িয়ে আপনি কমান্ডটা দিলেন সেটা কী কী অপশান নিয়ে চালানো হয়েছে। ম্যান পড়ে দেখুন, গাদা গাদা অপশান আছে।

১৬।। গ্লব এক্সপ্রেশন

অতিচিহ্ন বা মেটাক্যারেকটার এইসব চমকেবল রাশভারি নাম দিয়ে যাদের বোঝানো হচ্ছে সেইসব ভ্যাবলাকান্ত চিহ্নগুলোকে আমরা ব্যবহার করে আসছি সেই গোড়া থেকেই, এখানে একটা ওখানে একটা এই ভাবে ব্যবহার করে এসেছি। '>', '|', '\', '&', '&&', '*', '.' ইত্যাদি। এবার তাদের একটু গুছিয়ে চিনব। এদের মধ্যে একটা গুরুত্বপূর্ণ অংশ হল পাইকিরি ছক বা গ্লব এক্সপ্রেশন। একটা একটা করে চুন-চুনকে মারুঙ্গার গপ্পে যখন আর বিশ্বাস করা যায়না, গণহত্যাপন্থী হয়ে ওঠার প্রয়োজন পড়ে, তার জন্যেও ব্যবস্থা করা আছে আপনার সাধের গ্নু-লিনাক্সের সাধের ব্যাশে। অন্য আর এক নামেও এদের বোঝানো হয়, আগেই বলেছি, রেগেক্সপ বা রেগুলার এক্সপ্রেশন (RegExp)। খুচরো নড়বড়ে ব্যবহারকারীর কাঁপা কাঁপা ভিতু রকমে একটা একটা করে ফাইল খোঁজা নয়, এক লপ্তে লাটকে লাট ফাইল পাইকিরি রকমে নাড়াচাড়া করার জন্যে লাগে এই পাইকিরি ছক বা গ্লব এক্সপ্রেশন।

ভালো করে বোঝার জন্যে প্রথমে একটা এলেবেলে ডিরেক্টরি বানিয়ে নিন, তাতে ফাইল রাখুন নিচের তালিকামত। সব ফাইলই হল মন্ত্রী-উবাচের মত, মানে ফাঁপা এবং ফাঁকা। এরকম ফাইল বানিয়ে নিন লাটে। আপনি টাচউড, আগেই বলেছি, টাচ করুন আর ফাইল পয়দা হয়ে যাবে। এরকম নামগুলো ইচ্ছে করেই দিচ্ছি আমরা, যাতে পাইকিরি ছকের ব্যাকরণটা মালুম করা যায়। আপনি বিশ্বাস করবেন কিনা জানিনা, আমি আমার ফাইলের এত খারাপ নাম দিই-না, এত খারাপ ফাইলের নাম দেওয়া যায়না, কিন্তু এখানে এরকম নাম দেওয়ার কিছু কারণ আছে, এখুনি দেখতে পাবেন। সেই কারণগুলোকে মিলিয়েই হয়ত এর চেয়ে কম কদর্য নাম দেওয়া যেত, কিন্তু আমার মুন্ডুর মোট রসদ গত পাঁচমাস এই লেখায় রয়েছে, কল্পনাশক্তি ইত্যাদি অন্য প্রসেসগুলো সব নতুন পড়া গরমে জিভ বার করে ধুঁকছে। এখানে ফাইল মোট চব্বিশটা। প্রবন্ধ ষোলটা, ইংরিজির তিন, ভূগোলের চার, ইতিহাসের তিন আর অঙ্কের ছয়খানা। চারটে বিষয়ের চারটে বিবলিওগ্রাফি এবং চারটে নোটস। তাও তো, শুধু ছোট হাতের নাম দিয়েছি, বড় হাতের অক্ষর ব্যবহারই করিনি, জাটিল্য তাতে আরো বাড়ত।

essay.english.1.text	essay.geo.4.text	essay.math.3.text	essay.history.biblio
essay.english.2.text	essay.hist.1.text	essay.math.4.text	essay.math.biblio
essay.english.3.text	essay.hist.2.text	essay.math.5.text	notes.english.text
essay.geo.1.text	essay.hist.3.text	essay.math.6.text	notes.geography.text
essay.geo.2.text	essay.math.1.text	essay.english.biblio	notes.history.text
essay.geo.3.text	essay.math.2.text	essay.geography.biblio	notes.math.text

ধরুন এটা করছি হোম ডিরেক্টরির মধ্যে 'essay' বলে একটা সাবডিরেক্টরিতে। এবার এই '~/essay' ডিরেক্টরিতে দাঁড়িয়ে, '`ls`' মারলে, এই ফাইলগুলোই দেখতে পাব আমরা। বা '`ls *`' মারলেও তাই হবে, আমরা জানি, '`*`' চিহ্নটার মানে সব কিছু সমস্ত কিছু। শুধু ডটানন ফাইলদের ধরতে পারেনা চিহ্নটা, মনে পড়ছে, ছয় নম্বর দিনের আমাদের আলোচনা? কিন্তু যদি আমরা কমান্ড দিই, '`ls essay.[gh]*.text`', তাহলে আমরা এই ব্রিলিয়ান্ট নামের ফাইলগুলোর গোটা তালিকাটা আর দেখতে পাব না, হায়, দেখব শুধু ভূগোল আর ইতিহাস, 'geo' আর 'hist' সংক্রান্ত প্রবন্ধাবলীর তালিকা। ভূগোলের চার আর ইতিহাসের তিন পিস। এই কাঠামোটাই একটা পাইকিরি ছক বা গ্লব এক্সপ্রেশন।

essay.geo.1.text	essay.hist.1.text
essay.geo.2.text	essay.hist.2.text
essay.geo.3.text	essay.hist.3.text
essay.geo.4.text	

মানে, এই ছকটাকে ব্যাশ কী ভাবে বুঝছে সেই গোটাটা ভাবুন। প্রথমে মেলাচ্ছে সেই সমস্ত ফাইল যাদের শুরুতেই আছে পাইকিরি ছকের গোড়ার অংশটা, '`essay.`'। বিন্দুটাকে দেখতে আমি কখনোই ভুল করিনা, প্রথমত, আমার মা-র ডাকনাম পুঁটকি, আর দ্বিতীয়ত, বিন্দুতে যে কী বীভৎস সব কেলো ঘটে যায় তা ইউক্লিডের শ্বশুরও তেমন জানতেন না, গ্নু-লিনাক্স করতে গিয়ে আপনি যেমন জানবেন। তার মানে, ব্যাশ তুলে নিচ্ছে মোট কুড়িটা ফাইল, এবং বাদ দিচ্ছে চারটে, কারণ তারা '`essay.`' দিয়ে শুরু নয়। তারপরেই ব্যাশ যাচ্ছে ছকের দ্বিতীয় অংশটায়, দেখছে চৌকো ব্রাকেটের ভিতরকার অক্ষরগুলো। তুলে নেওয়া কুড়িটা ফাইলের মধ্যে যারা '`g`' বা '`h`' দিয়ে শুরু সবাইকে সে

তুলে নিচ্ছে তার হোলসেল ঝোলায়। তার মানে এই ধাপে বাদ পড়ে যাচ্ছে ইংরিজির তিনটে প্রবন্ধ একটা বিবলিওগ্রাফি মোট চারটে, এবং অঙ্কের মোট সাতটা ফাইল, কারণ তারা ‘g’ বা ‘h’ দিয়ে শুরু নয়। মানে, মোট এগারোটা ফাইল বাদ যাচ্ছে। কুড়ি থেকে এগারো গেল, পড়ে রইল নয়। কিন্তু কাজ এখনো শেষ হয়নি। এরপরে এই তালিকা থেকে সেই ফাইলগুলো ব্যাশ বাদ দিয়ে দিচ্ছে যাদের শেষে আবার সবিন্দু ওই ‘.text’ অংশটা নেই। এই ধাপে দেখুন বাদ পড়ছে দুটো ফাইল, ‘essay.geography.biblio’ এবং ‘essay.history.biblio’। কারণ, এদের শেষে পাইকিরি ছকের শেষ অংশটা, ‘.text’, নেই। তার মানে নটা থেকে দুটো বাদ গেলে পড়ে থাকছে ওই সাতটা ফাইল, এইমাত্র টেবিল করে যাদের আমরা দেখালাম। এবার বলুন তো আগের পাইকিরি ছক বা গ্লব এক্সপ্রেশনটা স্লাইট একটু বদলে যদি এরকম কমান্ড দিই, ‘ls essay.[^gh]*.text’, এতে ব্যাশ কোন ফাইলগুলোকে দেখাবে? এখানে এই ‘^’ চিহ্নটা ব্যাশকে বলে দিচ্ছে শুধু সেই অক্ষরেরা যারা ‘g’ বা ‘h’ নয়। ফাইলের তালিকাটা আমি দিয়ে দিচ্ছি, বুঝে নিন আপনি নিজে, এইমাত্র করা আলোচনাটার সঙ্গে মিলিয়ে। তফাত শুধু একটাই, ‘g’ বা ‘h’ এই সমাহারটার বদলে সেই সমাহার যা তৈরি তাদের নিয়ে যারা ‘g’-নয় এবং ‘h’-নয়।

essay.english.1.text	essay.math.1.text
essay.english.2.text	essay.math.2.text
essay.english.3.text	essay.math.3.text
	essay.math.4.text
	essay.math.5.text
	essay.math.6.text

এখানে যেমন আমরা চৌকো ব্রাকেটের মধ্যে সেই সব অক্ষরগুলো দিয়ে দিলাম যাদের দিয়ে শুরু করাতে চাই আমরা ব্যাশকে, সেরকম যদি অনেকগুলো অক্ষরগুলো হত, তাহলে সেটাকে একটা ধারাবাহিকতা হিশেবেও দিতে পারতাম। ধরুন, আমরা শুধু ইংরিজির ভূগোলের আর ইতিহাসের প্রবন্ধ কটাই দেখতে চাইছি, এই দশটা ফাইল, তখন আমরা চাইব শুধু ‘essay.[e-h]*.text’ ফাইলগুলোকে। এর মানে মধ্যের অংশটায় ‘e’ থেকে ‘h’ অব্দি যে কোনো বর্ণ দিয়ে শুরু থাকলেই তাকে অন্তর্ভুক্ত করবে এই ছকটা। এটা অন্য আর এক ভাবেও করা যেত, পাইকিরি ছক যদি দেন ‘essay.{english,geo,hist}.text’ সেটাও ধরবে সেই সমস্ত প্রবন্ধদের যাদের মধ্যের অংশটায় আছে, ‘english’ বা ‘geo’ বা ‘hist’। এই সংক্রান্ত আরো কোটি কোটি ডিটেইলস আছে, তার অনেকগুলোই ভারি মজার। ম্যান পড়ার নেশা করো ম্যান ... ইত্যাদি।

শুধু অঙ্কের প্রবন্ধের ফাইল কটাকে চাই যদি, আমার পাইকিরি ছকটা হবে, ‘essay.math.[1-6].text’। এর বদলে, যদি আমরা পাইকিরি বাজার করার জন্যে ছক দিতাম, ‘essay.math.?.text’ ছকও দিতে পারতাম। তাতেও একই কাজ করত। কিন্তু ‘?’ যেহেতু একটা কোনো বর্ণ বা অঙ্ককে বোঝায়, এখানে যদি দু-অঙ্কের কোনো সংখ্যা হয়, তাকে আর তুলবে না ব্যাশ। ধরুন, আপনার এলাকায় আপনি একটা জনগনতান্ত্রিক অঙ্কবিপ্লবের বিপ্লবী ভ্যানগার্ডদের দলে আছেন, এবং অঙ্কের প্রবন্ধ লিখেই চলেছেন, ঘ্যাশাঘ্যাশ ঘ্যাশাঘ্যাশ, শেষতমটা চলছে ‘essay.math.96.text’। এই অবস্থায় গোটাটাকে আনতে হবে দু-অঙ্কের জায়গা বানিয়ে, মানে ‘essay.math.??.text’। তার চেয়ে ভালো হয় যদি পাইকিরি ছকটা দেন ‘essay.math.*.text’, এতে করে আপনি হাজার লাখ কোটি প্রবন্ধ ছড়াতে থাকলেও ব্যাশের কোনো অসুবিধে হবেনা। কিন্তু সাবধান, যদি ‘essay.math.*’ দিয়ে ছেড়ে দেন, তাতে ‘essay.math.’ দিয়ে শুরু সব ফাইলকেই টেনে আনবে ব্যাশ। মানে অঙ্কের বিবলিওগ্রাফি নোটস সবকিছুই। এরকম আরো অজস্র ছক হয়, দুনিয়াটা তো নিয়ন্ত্রণ করছেই পাইকাররা, আমরা খুচরোরা নিতান্তই খুচরো। খুচরো মানুষের ইগো-ক্রাইসিস থেকে বাঁচতে গ্নু-লিনাক্স ব্যাশের গ্লব ব্যবহারের অভ্যেশ করার প্রেসক্রিপশন খুবই হচ্ছে আজকাল, এতে কাউন্সেলিং করানোর খরচ কমে আসছে।

এখানে কমান্ড লাইনে পাইকিরি ছক ব্যবহারের বিষয়ে একটা কথা বলে নিই। ধরুন আপনি একটা অঙ্কের প্রবন্ধের নাম যদি বদলাতে চান, ‘mv essay.math.1.text math.1’, ব্যাশ বিনা আপত্তিতে আপনার অঙ্কের প্রবন্ধটার সামনে আর পিছন থেকে ‘essay’ আর ‘text’ অংশদুটো উড়িয়ে দেবে। কিন্তু এবার এই কাজটা লাটে করতে যান, পাইকিরি ছকে, ‘mv essay.math.*.text math.*’, দিয়ে দেখুন, ব্যাশ রাজি হবেনা, অস্বীকার করবে, প্রমাদ বার্তা পাঠাবে স্ক্রিনে। কিন্তু কপি করা যাবে দিব্য। আপনি একটা ডিরেক্টরি বানান, ‘mkdir math’ দিয়ে।

এবার সেখানে সব কটা অঙ্কের প্রবন্ধ কপি করে দিন, ‘cp essay.math.*.text math/’, দেখুন, চোখের পলক না ফেলতেই হয়ে গেল, কোনো আপত্তিই করল না ব্যাশ। ‘ls math/’ মেরে আপনি দেখেও নিলেন। এ কেমন ধাঁচের বিদঘুটেপনা? কেন একটা করছে, অন্যটা করছে না? আসলে ‘mv’ কমান্ডটার আভ্যন্তরীন কাঠামোটা খেয়াল করুন। কমান্ডটার দুটো অংশ থাকে। ‘mv <name1> <name2>’ হল তার আকার। আমরা একটা ফাইলকে নড়াই অন্য একটা ফাইলে বা ডিরেক্টরিতে। যাকে নাড়াচ্ছি তার নাম ‘name1’, আর যেখানে নাড়াচ্ছি, তার নাম ‘name2’। যখন ‘name2’ হল একটা ডিরেক্টরির নাম, মানে, নড়িয়ে আমরা একটা ডিরেক্টরিতে রাখছি, সেটাই ভৌত অর্থে নাড়ানো। আর ‘name2’ যখন একটা ফাইলনাম, তার মানে আমরা নড়িয়ে অন্য ফাইলে, মানে অন্য নামের ফাইলে রাখছি, মানে ফাইলটার নাম বদলাচ্ছি। ‘mv’ কমান্ডটা এর বাইরে কোনো কাঠামোর আদেশ নিয়ে কাজ করতে পারছেনা। এবার মজার কথা, আমাদের এইমাত্র দেওয়া আপাতদৃষ্টিতে তো এই কাঠামোটা মেনে নিচ্ছে। তাহলে কেলোটা কোথায়? কেলোটা রয়েছে ব্যাশের ক্রিয়ার মধ্যে। আপনি যখন পাইকিরি ছক দিচ্ছেন, ব্যাশ সেটাকে বাড়িয়ে নিচ্ছে, এক্সপ্যান্ড করে নিচ্ছে। আপনার দেওয়া ‘mv’ কমান্ডের প্রথম অংশের পাইকিরি ছকটা বেড়ে গিয়ে হয়ে দাঁড়াচ্ছে ছ পিস খুচরো ফাইল। বুঝতে পারছেন, ‘mv’ কমান্ডের কাঠামোটাকেই আর খুঁজে পাচ্ছে না ব্যাশ। তাই কাজও করছে না। কিন্তু ‘cp’ কমান্ডের বেলায় সেরকম কোনো সমস্যা নেই, ছ খানা ফাইলের পরে আসছে তাদের পাঠানোর ঠিকানা, তাই ব্যাশের কাজ করতেও কোনো অসুবিধে হচ্ছে না।

কিন্তু ফাইলের নাম কি তাহলে লাটে বদলানো যাবেনা? ফাইল বদলানোর কাজটা খুচরো, নিতান্তই খুচরো। এই রে, বদলই তো জীবন, বেঁচে থাকা। আবার তো ইগো ক্রাইসিস? না, উপায় আছে, যদি বাড়াতে চান, সেটা ভারি সহজ, আগে নামগুলো ছোট করে নি, তারপর সেই ছোট হওয়া নামটা বাড়িয়ে দেখাচ্ছি। আগে ছোট করাটা দেখাই, সেটায় একটু মনোযোগ দিতে হবে। প্রথমে একটা ভ্যারিয়েবল বানান। আপনার একান্ত নিজস্ব একপিস লোকাল জিনিষ, যা আমরা কোটি কোটি বার করেছি।

```
filename=essay.math.text
```

একবার প্রতিধবনি মেরে দেখেও নিতে পারেন, মালটা কী দেখাচ্ছে। জানি ব্যাশ ভুল করেনা, তবু দেখে নিতে দোষ কী? আপনার তো দিতে কোনো ভুল হতে পারে। সেই পিলারের একটা এল-এর মত, তাতেও পিলার খাড়ায় ঠিকই, কিন্তু দুটো দেওয়াই ভালো, একটু পোক্ত হয়। এবার একটা কমান্ড দিন

```
echo ${name##essay.}
```

দেখুন, কমান্ড প্রম্পটে ফুটে উঠেছে, ‘math.text’। মানে পাঁঠার মুণ্ডটা বলি করে দিয়েছে, গোড়ার ‘essay.’ অংশটা। কিন্তু আমার পূজ্যপাদ অভিভাবকবৃন্দ যে কাজে আমার দক্ষতা অনস্বীকার্য বলে স্বীকার করে নেন, সেইরকম যদি আর কেউ হয়, সেও তার নিজের পাঠা লেজে কাটতে চায়, ‘.text’ অংশটা ওড়াতে চায়, তাহলে? আবার একটা কমান্ড দিন

```
echo ${name%%.text}
```

এবার দেখুন, কমান্ড প্রম্পটে ফুটে উঠেছে, ‘essay.math’। এই ম্যাজিকের মন্ত্রটা বুঝে নিন আগে। দুই জবাইয়ের দুরকম জবাফুল। শুরু ছাঁটার জন্যে ‘##’ আর শেষ ছাঁটার জন্যে ‘%%’। ‘##’ ব্যাশকে বলছে শুরু থেকে মানে বাঁ-দিক থেকে এই ‘essay.’ স্ট্রিং বা চিহ্নমালাটা অব্দি গোটাটা ছেঁটে ফেলো, আর ‘%%’ ব্যাশকে বলে দিচ্ছে, শেষ থেকে মানে ডান-দিক থেকে এই ‘.text’ স্ট্রিং অব্দি গোটাটা ছেঁটে ফেলতে। ভালো করে যদি জটিলতাটা বুঝতে চান, তাহলে খেজুরগাছের কাছে যান, এতবার যেতে যেতে কাঁটাগুলোও অনেক মসৃণ হয়ে গেছে এতদিনে। এবার, প্রকৌশলটা যখনই শিখে গেলেন এটাকে লাটে প্রয়োগ করুন পাইকিরি ছকে। আর ডিরেক্টরি তো আমাদের তৈরি আছে। এইমাত্র আমরা ‘math’ বলে ডিরেক্টরিটা বানিয়ে সেখানে কপি করে নিয়েছি অঙ্কের সমস্ত প্রবন্ধ। ‘math’ ডিরেক্টরিতে ঢুকে ‘ls’ কমান্ড দিয়ে দেখুন, ব্যাশ আপনাকে যাবতীয় ফাইলের তালিকা তুলে দিচ্ছে।

```
essay.math.1.text  essay.math.3.text  essay.math.5.text
essay.math.2.text  essay.math.4.text  essay.math.6.text
```

মানে, এই ডিরেক্টরির ফাইলগুলোয় গোড়ায় একটা ‘essay.’ দাড়ি, আর শেষে একটা ‘.text’ লেজ। প্রত্যেকটা রামছাগলের মত লম্বা ফাইলের দাড়ি আর লেজ ছেঁটে দিন। স্টেপ বাই স্টেপ এগোন। প্রথমে দাড়ি। পরে লেজ। কমান্ড দিন

```
for i in `ls *math*`; do mv $i ${i##essay.};done
```

এটা একটা ‘for’ লুপ। আগে একবার ধরে বোঝানো আছে, এখন দেখুন, প্রথমে ব্যাশ ‘ls *math*’ করে গোটা ডিরেক্টরিতে, মধ্যে ‘math’ অংশটা আছে এমন সব ফাইলের একটা তালিকা করে নিচ্ছে, এবার সেই তালিকাটাকে আমরা ডেকে নিয়েছি কমান্ডের মধ্যে কমান্ডে ব্যাককোট দিয়ে। এবার তার থেকে এক একটা করে ফাইল পড়ছে ব্যাশ এবং তাকে আমাদের এইমাত্র দেওয়া বলিপ্রক্রিয়ায় গোড়ার ‘essay.’ অংশটা ছেঁটে দিচ্ছে। এবার কমান্ড প্রম্পটে একবার ‘`ls`’ মারুন। দেখুন ফাইল দেখাচ্ছে

```
math.1.text    math.2.text    math.3.text    math.4.text    math.5.text
math.6.text
```

মানে, দাড়িছাঁটার কাজ ফিনিশ। এবার লেজ। কমান্ড দিন

```
for i in `ls *math*`; do mv $i ${i##essay.};done
```

কমান্ড প্রম্পট ফেরত এলে আর একবার ‘`ls`’ মারুন। এবার দেখুন, ছোট্ট ছোট্ট চিকণ ছাগশিশুর মত ফাইলে ভরে গেছে আপনার আদিকচক্রবাল ডিরেক্টরি। আহা তারুণ্যের কী স্লিমতা। কারুর নামে কোনো রেয়াজি মেদের কণাটুকু অব্দি নেই।

```
math.1  math.2  math.3  math.4  math.5  math.6
```

যা খাটলাম আপনাদের জন্যে যথেষ্ট। পাইকিরি ছক নিয়ে আর নয়। এবার আমরা ধরব অন্য মেটাক্যারেকটার বা অতিচিহ্ন। তার পর একটু লুপের দিকে আলগা লুক দেওয়া। তারপরেই ছুটি। ও, নাম ছোট করা তো হল, এবার এদের একবার বড় করে নেওয়া যাক। তার জন্যে কমান্ডটা খুবই সহজ, এবং একটা কমান্ডেই এই ছাগলছানাদের দাড়ি এবং লেজ সহ পরিপূর্ণতায় পৌঁছে দেওয়া যাবে।

```
for i in *; do mv $i essay.$i.text; done
```

এবার ‘`ls`’ মারলেই দেখতে পাবেন, ফাইলগুলো আবার তাদের যাত্রাশুরুর এক নম্বর চৌখুপিতে পৌঁছে গেছে। এই যে বারবার ‘`do`’ আর ‘`done`’ দিয়ে করছি, এটা কিন্তু আদৌ জরুরি নয়, আসলে লুপের স্ক্রিপ্টোচিত কাঠামোর সঙ্গে চেনাশোনাটা বাড়িয়ে রাখতে চাইছি। আর একটা ছোট উদাহরণ দিয়ে রাখি এখানে, যা স্ক্রিপ্ট বানাতে গিয়ে বেশ কাজে লাগে। পথ-সহ একটা ফাইলনামের গোটাটার থেকে তার নিছক ফাইলনাম আর পথ — এই দুটোকে আলাদা করে বার করে নিতে পারে ব্যাশ, ‘dirname’ আর ‘basename’ কমান্ড দুটো দিয়ে। ধরুন, পথ বা ঠিকানা সহ একটা ফাইল, ‘/usr/local/mplayer/conf’। এটা দেখে আমরা সচেতন মানুষেরা নিজে নিজেই বুঝে নিতে পারি, ‘/usr/local/mplayer’ হল ফাইলটার ঠিকানা বা পথ মানে ‘dirname’, আর ‘conf’ হল নিছক ফাইলনাম মানে ‘basename’। এবার, ‘`dirname /usr/local/mplayer/conf`’ কমান্ড দিলে, ব্যাশ তার কমান্ড প্রম্পটে দেখাবে, ‘`/usr/local/mplayer`’। আর ‘`basename /usr/local/mplayer/conf`’ কমান্ড দিলে দেখাবে ‘`conf`’। এই দুটোর সুন্দর ব্যবহার আছে অনেক, কিন্তু প্রথমত সেগুলো অনেক পরিণত স্ক্রিপ্টে, আর সহজ কোনো উদাহরণ মাথায় ভেবে নিতে পারছি না। তার খুব একটা দরকারও নেই, এই অর্থে যে, আজকের এই গোটা আলোচনাটা আপনাকে এগিয়ে দিতে চাইছে স্ক্রিপ্ট লেখার আগেও স্ক্রিপ্ট পড়ার দিকে। সিস্টেমের মধ্যেই যে অজস্র স্ক্রিপ্ট এমনিতেই আছে, তারা কোথায় কখন ঠিক কী করে — সেই বোঝাটার জন্যেই দরকার পড়ে স্ক্রিপ্ট পড়া। আর এই স্ক্রিপ্টগুলো তাদের লেখা যারা সবচেয়ে ভালো করে এই কাজটা জানে। তাই, শেখার জন্যে তারাই আদর্শ। আজকের আলোচনাটা জাস্ট সেই শেখাটার দিকে পৌঁছে দেওয়া। এখন থেকে যাতে, কোনো স্ক্রিপ্ট সামনে দেখে আপনি আন্দাজ করার চেষ্টা শুরু করতে পারেন, অন্তত কোথায় কোথায় বুঝতে পারছেন না সেইটুকু বুঝতে পারেন। একটা নতুন জিনিষের বেলায় সেই শেখাটাই কিন্তু একটা বিরাট ব্যাপার। এই পাইকিরি ছকের কাজ শেষ। এবার এগোনো যাক অন্যান্য অতিচিহ্নের দিকে।

১৭।। কয়েকটা মেটাক্যারেকটার বা অতিচিহ্ন

এই অতিচিহ্ন বা মেটাক্যারেকটারদের এই নামে ডাকার কারণ এই যে, এরা নিজেরা নিজেরা যেমন নিছক একটা চিহ্ন, তেমনি ব্যাশের কাছে এদের বিশেষ কিছু অর্থও আছে। কোন অবস্থায় এদের চিহ্ন হিশেবেই পড়বে ব্যাশ, কোন অবস্থায় সেই বিশেষ অর্থে তারও নিয়মকানুন আছে। সেটা আমরা এখুনি জানব।

শব্দযতি বা ওয়ার্ড সেপারেটর

সচরাচর একে আইএফএস বলে ডাকা হয় (IFS — **I**nternal-**F**ield-**S**eparator)। আমরা 'set' কমান্ড দিয়ে ব্যাশের এনভায়রনমেন্ট ভ্যারিয়েবলদের পড়তে শুরু করেছি সেই ছয় নম্বর দিন থেকেই। এবার সেই 'set' কমান্ডের ফলাফলকে একটু গ্রেপ মারা যাক। 'set| grep 'IFS''। এতে যে ফলাফল আপনার স্ক্রিনে দেখাবেন ব্যাশবাবু, তাতে এজাতীয় একটা লাইন পাবেনই, 'IFS=$' \t\n''। এই লাইনটার ডানদিকের অংশটা খেয়াল করুন। দুটো কোটের মধ্যে রয়েছে তিনটে জিনিষ। একটা ফাঁকা ভূমি। তারপর '\t' মানে ট্যাব চিহ্ন। তারপর একটা নিউলাইন '\n'। শূন্য নম্বর দিনে এবং তার পরেও অ্যাসকি কোড আলোচনার সূত্রে আমরা ছুঁয়ে এসেছি এগুলো, মনে করুন। এই তিনটে জিনিষের যে কোনোটাকেই ফিল্ড সেপারেটর বলে ধরে নেয় ব্যাশ। একটু আগে আমরা যখন 'set' কমান্ড দিয়ে অবস্থানগত নিয়ন্তাদের বানাচ্ছিলাম, মনে আছে, এক একটা করে স্পেস পেয়েছে ব্যাশ, আর এক একটা করে নতুন প্যারামিটার বানিয়ে নিয়েছে। ওখানে ট্যাব বা নিউলাইন হলেও একই হত। নিউলাইন মানে কিন্তু ক্যারেজ রিটার্ন নয়। ক্যারেজ রিটার্ন পাই আমরা এন্টার চাবি টিপে। মানে, এই আমার কমান্ড খতম, এবার বাছা ব্যাশ তুমি পিন্ডি চটকাতে শুরু করো। এই নিউলাইন বা '\n' তৈরি হয় যখন আমরা একটা '\' দিয়ে তারপর এন্টার মারি। এই ভাবে আমরা খুব লম্বা কমান্ডকে এক লাইন থেকে আর একটা লাইনে টেনে নিয়ে যেতে পারি, তাতে কমান্ড সমাপ্ত হয়না। এমনিতেও ব্যাশ স্ক্রিনের চওড়ায় না ধরলে স্ক্রিনে পরের লাইনে নিয়ে যায়, কিন্তু তা আসলে স্ক্রিনের নতুন লাইন, কিন্তু ব্যাশের কাছে নয়, উপরের লাইনটা শেষ হয়ে যেতে পারে যে কোনো শব্দের মাঝখানে, এই '\' দিয়ে তারপর এন্টার মেরে আপনি আপনার পছন্দমত জায়গায় ভাঙতে পারেন কমান্ডটাকে। শব্দযতি বা ওয়ার্ড সেপারেটর নামটার চেয়েও ফিল্ড সেপারেটর বা জমির আল বললে ব্যাপারটা আরো ভালো বোঝা যায়। দুটো জমির মধ্যে সীমাটা স্লিম হতে পারে, একটা স্পেসের, ট্যাবমার্কের মত আদনান সামি সাইজের হতে পারে, আবার দুটো জমির মধ্যে বয়ে যাওয়া নিউলাইনের নদীও তো জমিদুটোকে বিশ্বস্ত ভাবেই আলাদা করে।

সারিবদ্ধ আদেশের সেমিকোলন ';'

ইনিও আমাদের চেনা। আমার গদ্য শেখার শুরু যার হাতে, সেই দান্তে, ভাস্করজ্যোতি সমাস করেছিল, দাঁত-অন্তে-যাহার, বহুব্রীহি, ছড়া চলত, দান্তের গাট্টা টাকায় আটটা, শিশিরবাবু স্যার বলেছিলেন, বাক্যটা শেষ করলেই করতে পারতিস, কিন্তু করলি না, একসাথে আরো কিছু বলে নিতে চাইছিস — এর মানে সেমিকোলন। ঠিক তাই ব্যাশের বেলাতেও। আলাদা ভাবে একটা কমান্ড দেওয়ার পরে এন্টার মেরে ব্যাশকে দিয়ে কাজ করিয়ে নিতে পারতাম, কিন্তু একসঙ্গে দিয়ে দিলাম একাধিক কমান্ড, এই জায়গায় আসে ';'। আমাদের ব্যবহার করা সেমিকোলন দেওয়া কমান্ডগুলো মনে করুন।

'&' মানে পর্দে-কে-পিছে

এই অ্যাম্পারস্যান্ড চিহ্ন '&' আমরা কাজে লাগিয়েছি সম্মুখভূমির সক্রিয় প্রক্রিয়া বা প্রসেসকে পশ্চাৎভূমিতে পর্দার পিছনে নিয়ে যাওয়ার এই চিহ্ন আমরা অনেকবার ব্যবহার করেছি। লিটারালি সেটাই লাগে, কনসোলের গম্ভীর কালো স্ক্রিনের পিছনে চলে গেল, কিন্তু উবে যায়নি, কারন 'fg' মারলেই ফেরত আসে।

সুজনদের তেঁতুলপাতা মানে ব্রাকেট, '(' এবং ')'

একলা নয়, একসাথে চলার জন্যে দরকার পড়ে। ঠিক মনে পড়ছে না, কিন্তু এত এত কমান্ড নিয়ে আলোচনা হয়েছে, একটাও ব্রাকেট আসেনি? কে জানে? নয় নম্বর দিনের ২.২ সেকশনে '/etc/sysconfig' আলোচনার সূত্রে আমরা 'finger' কমান্ডটার কথা বলেছিলাম, মনে আছে? কোনো একজন ব্যবহারকারী বিষয়ে তথ্যগুলো তুলে ধরে সামনে? এবার সেই কমান্ডকে আরো সুজনের সঙ্গে মিশিয়ে, ব্রাকেটের তেঁতুলপাতায় বসিয়ে, সময়ের সঙ্গে সঙ্গে পরিবর্তমান আত্মজ্ঞানের উৎস করে তোলা যাক। কমান্ড দিন

```
(finger `whoami`; date)|cat
```

আপনি অবশ্য একে রিডাইরেক্ট করে একটা ফাইলেও রেখে দিতে পারতেন, আপনার আত্মজীবনীর একটা অংশ হিশেবে। আমার আত্মজীবনীর একটা ছেঁড়াপাতা আপনাদের সামনে তুলে ধরা যাক। আমার আলস্যর কথা কেমন হিংস্র রকমে ঘোষণা করে দিয়েছে দেখুন, বিশেষত পরপর 'No Mail. No Plan.' অংশটায় গামা প্যাথোজ যেন থৈ থৈ করছে — নো ওয়ান রাইটস টু দি কর্নেল।

```
Login: dd                                Name: dipankar das
Directory: /home/dd                      Shell: /bin/bash
On since Sat Mar 20 09:08 (IST) on tty1,  idle 0:18
No Mail.
No Plan.
Sun Mar 21 09:26:52 IST 2004
```

ব্রেস বা '{' এবং '}' চিহ্নও একটু আগে আমরা ব্যবহার করেছি, ব্যাশের পাইকিরি ছক বোঝার সময়। আরো অনেক খ্যাঁচসঙ্কুল ব্যবহার আছে এর। ম্যানুয়াল পড়ে জেনে নিন। ব্যাশের পাইকিরি ছকে আরো কী কী অতিচিহ্ন ব্যবহার করে এসেছি দেখুন তো সেটা মনে পড়ছে কিনা? '*', '?', '[' , এবং ']'। এর সঙ্গে আছে রিডাইরেকশন এবং পাইপিং-এর চিহ্ন, মানে '>', '>>', '<' এবং '|'। শব্দযতির কথা বলতে গিয়ে এল লাইন বাড়ানোর চিহ্ন '\'। আছে আদেশ বদল বা কমান্ড সাবস্টিটিউশনের চিহ্ন '`'। বা '$' যা আদেশ বদলও করে, আবার ভ্যারিয়েবলের মানও বোঝায়। আরো আছে, মনে করুন, যে ডিরেক্টরিতে আছেন, সেই মুহূর্তে সেই ডিরেক্টরি বোঝাতে '.'। এই ডিরেক্টরিটা যে ডিরেক্টরির মধ্যে আছে, মানে এর উপরের ডিরেক্টরি বোঝাতে '..'। হোম ডিরেক্টরি বোঝাতে '~'। আর একটা অতিচিহ্ন তো আমরা সেই এক নম্বর দিন থেকেই দেখে আসছি, মানববোধ্য মন্তব্য বা কমেন্টস বোঝাতে, '#', লাইনের শুরুতে যে চিহ্নটা থাকে মানেই সিস্টেম বুঝে যায় এটা মানুষের পড়ার জন্যে। আরো বোধহয় কিছু আছে, ভুলে যাচ্ছি। যাকগে ব্যাশ ম্যানুয়ালের এরকম কোনো ভুল হবেনা। এখন একটু পড়ে দেখে নেওয়া যায়, কিন্তু আমায় দিয়ে সিম্পলি আর হচ্ছেনা। এখন এটা শেষ হলে বাঁচি। দাঁড়ান আর একটা, তাও নিজের মনে পড়েনি, পাতা ওল্টাতে গিয়ে চোখে পড়ল।

শর্তাধীন ক্রিয়ার আদেশ '&&' এবং '||'

ব্যাশকে শর্তাধীন ক্রিয়ার বা কনডিশনাল একজিকিউশনের আদেশ দেওয়ার জন্যে লাগে এই ডাবল অ্যাম্পারস্যান্ড, '&&', এবং ডাবল পাইপ, '||' — মানে বিশেষ একটা কাজ তবেই করবে যদি একটা বিশেষ শর্ত মেটে। এর মধ্যে প্রথমটা আপনাদের চেনা। আগের কাজ শেষ হলে তবে পরেরটায় যাবে, মনে করে দেখুন, আপনার মুন্ডু প্যাকেজ ইনস্টলেশনের সময়ে আমরা কমান্ড দিয়েছিলাম, './configure && make && make install'। যাতে আগের কাজ ঠিক ভাবে সমাপ্ত হওয়ার আগেই পরের কাজে চলে না-যায়। তখন জানতাম না, কিন্তু এখন জানি, একটু টেকনিকাল ভাষায় বলতে গেলে, '&&' চিহ্ন ব্যাশকে জানায় শুধু তখনই পরবর্তী আদেশে যেতে যদি আগের আদেশের নিষ্ক্রমণ স্থিতি বা একজিট স্ট্যাটাস শূন্য হয়। সাফল্যকে শূন্য ভাবে ব্যাশ, মনে আছে তো? ঠিক এর উল্টোটা হল '||'। এই '||' চিহ্নের ডানদিকের আদেশ শুধু তখনই পালন করবে ব্যাশ যদি আগের আদেশটা ব্যর্থ হয়ে থাকে, মানে তার নিষ্ক্রমণ অবস্থা শূন্য না-হয়। ধরুন আপনি জানতে চাইছেন আপনার 'math.text' ফাইলে 'integer' শব্দটা আছে কিনা। আমরা জানি এর কমান্ড 'grep', সেই লাইনগুলোকে নিংড়ে আনে যাতে এই 'integer' শব্দটা আছে এবং স্ক্রিনে দেখিয়ে দেয় লাইনগুলো। আর যদি শব্দটা একবারো না পায়, তাহলে সেই নির্বাক নিঃশব্দ প্রত্যাগমন। শূন্য কমান্ড প্রম্পট খাঁখাঁ করছে স্ফটিক অন্ধকারের দেওয়ালে। তা না করে যদি অন্তত কিছু চান স্ক্রিনে, তাহলে কমান্ড দিন

```
grep 'integer' math.text || echo NO 'integer' found
```

এবার, যদি 'grep' পেয়ে যায় 'integer' শব্দটা তাহলে তো লাইনগুলোই দেখাবে, আর যদি না-পায় তখন দেখাবে তার বার্তা, 'NO 'integer' found'। এবার মনে হচ্ছে অতিচিহ্ন মোটামুটি নামল। কিন্তু এদের নিয়ে এখনো একটা কুচো সমস্যা আছে। এই অতিচিহ্ন বা মেটাক্যারেকটারগুলো তো বললাম, বিশেষ কিছু অর্থ বহন করে আনে ব্যাশের কাছে, বিশেষ কিছু আদেশ। কিন্তু ধরুন আপনি, কখনো, ওই বিশেষ অর্থে নয়, নিছক একটা চিহ্ন হিশেবেই

ওই বিশেষ মেটাক্যারেকটারটাকে ব্যবহার করতে চান, তখন কী করে করবেন? ধরুন আপনি ব্যাশকে কমান্ড প্রম্পটে দিলেন, 'echo An asterix is a *'। আপনার কী মনে হচ্ছে? খুব সাধারণ একটা লাইন তো, ইকো করেই দেখাবে ব্যাশ। ভুল। করে দেখুন, '*' চিহ্নটার আগের অংশটা, মানে 'An asterix is a' অংশটা একই থাকবে, কিন্তু তারপরে আসবে, যে ডিরেক্টরিতে দাঁড়িয়ে কমান্ডটা দিয়েছেন, সেই ডিরেক্টরির ভিতরের মোট ফাইলের তালিকা। কারণ, ব্যাশের কাছে '*' মানেই তো তাই। তাহলে এই অতিচিহ্নদের হাত থেকে মুক্তির উপায় কী? কখন একটা '*'-কে '*' বলেই বোঝানো যাবে?

১৮।। অতিচিহ্নের থেকে নিস্তার

অতিচিহ্নের অতি-আচার থেকে রেহাই পাওয়ার উপায় একদম বিষে বিষে বিষক্ষয়, মানে আরো কিছু অতিচিহ্ন। দাঁড়ান, সেটার আগে, এই বিন্দু বা '.' চিহ্ন খেয়াল রাখাটা কতটা জরুরি হয়ে পড়ে, একটু আগে বলছিলাম না, তার এইমাত্র দেখা একটা উদাহরণ বলি। আপনাদের জন্যে এটা লেখার আগে আমি একবার করে দেখতে গেলাম। গিয়ে কমান্ড দিলাম 'echo An asterix is a *.'। আরে, আমার বোধভাষ্যি মত, এখানে তো ফাইল তালিকা দেখানোর কথা, ব্যাশ তাই করে বলেই এতদিন জানি। কিন্তু তা না-করে, ব্যাশ হুবহু একদম পুরো কথাটা ইকো করছে। কী হল? তারপর দেখি, ও হরি, আমি '*' না দিয়ে দিয়েছি '*.', তাই ব্যাশ সেখানে একে বাড়িয়ে তোলার এক্সপ্যান্ড করার মত কোনো ফাইল তালিকাই পাচ্ছে না, কারণ যেখানে দাঁড়িয়ে দিয়েছি কমান্ডটা সেখানে এমন কোনো ফাইলই নেই যা একটা বিন্দুতে শেষ হচ্ছে। ইন ফ্যাক্ট এরকম কোনো ফাইল আমি আজতক দেখিনি। তাই এক্সপ্যান্ড করতে না-পেরে ব্যাশ চিহ্নটাকে একদম আক্ষরিক আকারেই লিখে দিয়েছে।

এরকম ফাঁকতালে সমাধান বাদ দিন, সত্যিকারের সমাধানের একাধিক উপায় আছে এখানে। এক ব্যাকস্ল্যাশ বা '\'। আর দুই, কোট চিহ্ন বা উর্দ্ধকমা। কোট কিন্তু আবার তিনরকম। এক ব্যাককোট, '`', দুই সিঙ্গল কোট ''', আর তিন, ডাবল কোট, '"'। এর মধ্যে ব্যাক কোটকে তো আমরা আলাদা করে নিয়েছি আগেই, আদেশ বদল বা কমান্ড সাবস্টিটিউশনের চিহ্ন হিশেবে। এবার এই ব্যাকস্ল্যাশ, এককোট আর জোড়াকোট — '\', ''', '"' — এই তিনটে অতিচিহ্ন ব্যবহার করা হয় অতিচিহ্ন থেকে বাঁচার অতিচিহ্ন হিশেবে। কিন্তু তাদের একজনের থেকে আর এক জনের ব্যবহারের পার্থক্য আছে।

ধরুন, আমাদের ওই তারকাচিহ্ন চেনানোর লাইনটা ভাবুন। 'echo An asterix is a *'। একে যদি হুবহু আমরা স্ক্রিনে দেখতে চাই। এই তিনটে উপায়ের তিনটেই সমান কার্যকরী।

```
echo An asterix is a \*
echo 'An asterix is a *'
echo "An asterix is a *"
```

উপরের তিনটের যে কোনোটাই স্ক্রিনে ফুটিয়ে তুলবে সেই অভিজ্ঞান, 'An asterix is a *'। মানে, ব্যাশ যেখানে '*' চিহ্নটাকে অতিচিহ্ন নয়, স্বাভাবিক একটা সাধারণ চিহ্নের মতই পড়বে। অন্য একটা চিহ্নমালা '*?*' দিয়ে এটা করে দেখুন। '\' বেলায় দিতে হবে '*\?*'। ''' হলে ''*?*''। আর '"' চিহ্ন হলে '"*?*"'। এই তিনটে কেসের প্রত্যেকটাতেই আপনি ফল পাবেন এক, ব্যাশ ফুটিয়ে তুলবে '*?*'। তাহলে এদের তফাতটা কোথায়? তফাত মালুম করার জন্যে আমাদের পুরোনো চেনা তক্তা '$PATH' আছে। তাতে একবার করে এই তিনটে পেরেকই মেরে দেখুন। তিনবার এই তিনটে কমান্ড দিন। বোঝা যাবে, এই '$' চিহ্নটাকে ভ্যারিয়েবলের মানের সঙ্কেত হিশেবে ভেবে ব্যাশ এটাকে পথনির্দেশের সিস্টেম ভ্যারিয়েবল বলে ভাবছে, তাই তার মানটাকে স্ক্রিনে ফুটিয়ে তুলছে? নাকি, হুবহু গোটাপৃথিবীর স্বপ্নের সবুজের ছবি নিয়ে একটা আক্ষরিক বর্ণসমারোহ করে দেখাচ্ছে? ডলার ছাড়া স্বপ্ন আর কই? ফ্যান্টাসি?

```
echo \$PATH
echo '$PATH'
echo "$PATH"
```

এর প্রথম দুটোয় ব্যাশ ভাববে একদম হুবহু একটা চিহ্ন হিশেবেই। মানে স্ক্রিনে ফুটিয়ে তুলবে, ‘$PATH’। কিন্তু তৃতীয়টার বেলায় যা পাব তা আমাদের চেনা, আমার মেশিনে সুজে সিস্টেমে ‘dd’ নামের ব্যবহারকারীর ব্যাশের পথনির্দেশ।

```
/home/dd/bin:/usr/local/bin:/usr/bin:/usr/X11R6/bin:/bin:/usr/games:/opt/gn
ome2/bin:/opt/gnome/bin:/opt/kde3/bin:/usr/lib/java/jre/bin:/opt/gnome/bin
```

তার মানে বুঝতে পারছেন, জোড়াকোট বা ‘"’ চিহ্ন ‘$’ চিহ্নকে অতিচিহ্ন বা মেটাক্যারেকটার বলেই ভেবেছে, কিন্তু ব্যাকস্ল্যাশ, ‘\’, বা এককোট, ‘'’ তাকে অতিচিহ্নতা থেকে নিষ্কৃতি দিয়েছে। এই তিনটে চিহ্নের আর একটা আত্মীয়তার জায়গাও আছে। ব্যাকস্ল্যাশ চিহ্ন ‘\’ দিয়ে একটা লাইন থেকে পরের লাইনে কমান্ড টেনে নিয়ে যাওয়ার কথা বলেছি আগেই। এই চিহ্নের পরে এন্টার মারলে নতুন লাইনে যায় প্রম্পট, কিন্তু কমান্ড নেওয়াটা শেষ হয়না। ঠিক সেই একই কাজ করা যায় এককোট ‘'’ আর জোড়াকোট ‘"’ চিহ্ন দিয়েও। একটা এককোট বা জোড়াকোট শুরু করার পর যতক্ষণ না আপনি তার দোসর এককোট বা জোড়াকোট চিহ্নটা মারছেন, ততক্ষণ একটা কমান্ডই চলতে থাকবে। এন্টার মারলে প্রম্পট পরের লাইনে যাবে, কিন্তু কমান্ড শেষ হবেনা। এই তিনটে চিহ্নের তফাতটা একবার জেনে রাখা যাক। তবে আর একবার বলে রাখি, এগুলো ব্যাশ ম্যানুয়াল থেকে একবার মিলিয়েনিন, একদম নিখুঁত বলছি কিনা আমি নিজেই শিওর নই। মানে যেটুকু বলছি সেটা তো সিস্টেমে মিলিয়ে দেখেই বলছি, কিন্তু এর বাইরে আরো কিছু থাকতে পারে, যা আমি নিজেই জানিনা তো লিখব কী।

নিষ্কৃতিচিহ্ন	কাজ করার ধারা
ব্যাকস্ল্যাশ ‘\’	একটা অতিচিহ্নের এককে কাজ করে, যতগুলো অতিচিহ্নকে আমি আক্ষরিক ভাবে মানে সাধারণ চিহ্নের মত ব্যবহার করতে চাইছি তার প্রত্যেকটার জন্যে একটা করে দিতে হবে। মানে দুটো তারকাচিহ্নের জন্যে, ‘**’, দিতে হবে দুটো, ‘**’।
এককোট ‘'’	যতটা এলাকাকে দুটো এককোট চিহ্নের মধ্যে রাখা হবে, একটা শুরুর, একটা শেষের, তার মধ্যে গণতন্ত্র হি গণতন্ত্র, সবাই সমান, সবাই জাস্ট চিহ্ন, অতিচিহ্ন বলে তো কিছু হয়না ভাই।
জোড়াকোট ‘"’	নিষ্কৃতিচিহ্ন হিশেবে রীতিমত বুদ্ধিজীবী, আলাদা আলাদা জায়গায় আলাদা আলাদা বক্তব্য। যতটুকু জায়গাকে দুটো জোড়াকোটের মধ্যে রাখা হবে, শুরুর আর শেষের, সেখানে কিছু চিহ্নকে টেনে নামিয়ে আনবে মাটির পৃথিবীতে, একদম এক ভোটে জীবন, এক গুলিতে মৃত্যু, নরমাল চিহ্ন। কিন্তু তিনটে অতিচিহ্নকে অতিচিহ্নের ওজনেই রেখে দেবে। তারা হল ব্যাকস্ল্যাশ ‘\’, ডলার ‘$’, আর ব্যাককোট ‘`’।

মা তারা ব্রহ্মময়ী মা, একটু চান করে আসি। আর মাত্র একটা সেকশন। আজ রোববার, একুশে মার্চ বিকেল পাঁচটা ছেচল্লিশ, আর ঘন্টা দুয়েকের মধ্যে মালটা ফিনিশ হবে।

১৯।। নিয়ন্ত্রণ কাঠামো

আমাদের এই বইয়ের শেষ সেকশনে আবার আমরা ফিরে এলাম নিয়ন্ত্রণ কাঠামোর ধারণাটায়। এটাই তো তাই যা কম্পিউটারকে কম্পিউটার করেছে, প্রাককম্পিউটার সমস্ত হিশেবযন্ত্র থেকে আলাদা করে এনেছে। এর ভিতরেই উপস্থিত সচেতন মানুষ। তার সারোগেট, তার প্রতিনিধি — যার অন্য নাম প্রোগ্রাম। আজকের আলোচনার ১০ নম্বর সেকশনে ভন নয়মান কাঠামোর কেন্দ্রীয় নিয়ন্ত্রণ বা ‘CC’ অংশটাকে ভাবুন। এটাই তো তাই যা স্টোরড প্রোগ্রাম কম্পিউটারের আগে অব্দি অন্য মেশিনগুলোয় ছিলনা। তার আগে অব্দি শুধু র ডেটা বা কাঁচা তথ্যই থাকত মেশিনে। এখন থেকে ইন্সট্রাকশন ডেটা বা আদেশ তথ্যও থাকতে শুরু করল। তিন নম্বর দিনের কম্পিউটারের প্রাগ-ইতিহাসের আলোচনাটা মনে করুন। সেই ডিজিজড এঞ্জেল, ক্লিষ্ট দেবদূত, আডা লাভলেস, তার সেই স্বপ্নগুলো, যাকে আমরা এই শতাব্দীর সবচেয়ে উত্তেজক সায়েন্স ফিকশন বলে ডেকেছিলাম। কেন? মনে করুন সেই কার্ড গুলোর কথা। কার্ডগুলো সংখ্যাকে ধারণ করছে, হিশেবকে ধারণ করছে, কার্ড আর চাকা আর গিয়ার আর ডাণ্ডা। ডিফারেন্স ইঞ্জিন অব্দি। এল অ্যানালিটিকাল ইঞ্জিন, সেই নতুন ধরনের কার্ডের ধারণা, যা আর হিশেব বা সংখ্যা রাখেনা, রাখে অন্য

কার্ডের কাজের খতিয়ান, তার নিয়ন্ত্রণ। মেশিনের মধ্যেই ঢুকে এল অমেশিন, জড়ের মধ্যে জীবন, সচেতনতা। শূন্য নম্বর দিনের সেই পাঁচ ধরনের উপাদানের কথা মনে করুন। ইনপুট উপাদান, আউটপুট উপাদান, তথ্য প্রসেসিং বা চটকানোর উপাদান, আর তথ্য হোল্ডিং বা ধারণের উপাদান — এই তো গেল চারটে। পাঁচ নম্বরটা? ঠিক মানুষের হাতের তালুতে যা হল, পাঁচ নম্বর আঙুলটা অন্য চারটে আঙুলের সাধারণ সমতল থেকে বেরিয়ে এল, ঘুরে গেল কয়েক ডিগ্রি — ভেবে দেখুন — এই ঘুরে যাওয়া বেঁকে যাওয়া বেরিয়ে যাওয়া বুড়ো আঙুলটাই হল প্রকৃতির উপর অন্য প্রাণীদের থেকে আলাদাতর মানবিক নিয়ন্ত্রণের গোটা ইতিহাসটা। ঠিক তাই ওই পাঁচ নম্বর উপাদানটারও। অন্য চারটে উপাদানকে দেখুন। তারা কী করে? তথ্যকে নিয়ে নাড়াচাড়া করে। যা আসলে কম্পিউটারেরই কাজ। আর পাঁচ নম্বরটা কী করে? সে তথ্য নিয়ে কিছু করেনা। অন্য চারটে উপাদানের কাজকে নিয়ন্ত্রণ করে। এটাই ব্রেক। এটাই ভিশন। অলৌকিকতা। যা সবার থাকেনা, সকলেই কবি নয়, কেউ কেউ কবি। আডার ছিল।

এটাই নিয়ন্ত্রণ কাঠামো। মেশিনের মধ্যেই মেশিনের সেকেন্ড অর্ডার। মেশিনের বাহির মেশিনের অপর মানে মানুষ যখন ঢুকে গেল মেশিনের ভিতর। ঠিক কাজের নিরিখে ভাবা যাক। ধরুন একটা সহজতম ব্যাশস্ক্রিপ্ট। যেখানে নিছক একটা কমান্ডকে সরাসরি ল্যান্ড করিয়ে দেওয়া হয়েছে। 'trashcopy' নামে একটা ফাইল বানান। তাতে নিচের তিনটে লাইন টাইপ করে ঢোকান।

```
#!/bin/bash
cp /etc/trash ~/
echo "Done"
```

প্রথম লাইনটা তো যে ধ্রুবপদ দিয়েছো বাঁধি — ব্যাশপুজোর মন্তর। দ্বিতীয় লাইনটাই হল কাজ, জাস্ট '/etc' ডিরেক্টরির ভিতর একটা '`trash`' নামে ফাইল কপি করে আনা হোম ডিরেক্টরিতে। আর তৃতীয় লাইন হল ব্যাশবাণী, করেছি গো অ্যাক্কেরে করে ফেলেছি। এবার স্ক্রিপ্টটাকে চালনীয় বানান, '`chmod +x trashcopy`'। যেই এক্সিকিউটেবল হল, এবার চালান, '`./trashcopy`'। কী পেলেন?

```
cp: cannot stat `/etc/trash': No such file or directory
Done
```

প্রথম লাইনটা দেখুন, কপি করবে কী বেচারা, ওরকম কোনো নামের ফাইল ওই ডিরেক্টরিতে, কেউ বাপের জন্মে শোনেনি, ফাইলই নেই, তার কপি করবে কী? কিন্তু বুদ্ধির পরাকাষ্ঠাটা পরের লাইনে। সেই কান এঁটো করা হাসি সহ, '`Done`'। হায় গাধা, তুই তো কাজটা পারিসই নি, তাও '`Done`'। এবার ভাবুন, আসলে গাধাটা কে? আমরা তো ব্যাশকে না- গাধা হতে শেখাই নি। আডার সেই বাক্যটা মনে করুন, এমন সব কাজই করতে পারবে মেশিন, যা কী করে করতে হয় আমরা বলে দিতে পারব। আমরা তো শেখাইনি, কী করে ডিসিশন নিতে হয়, কোথায় শুরু করব, কোথায় থামব — এই প্রোপোরশন বোধও শেখাইনি, শেখাইনি কী করে নিজের ভুল সংশোধন করতে হয়। আমরা শেখাই নি, ফাইলটা আছে না নেই সেটাও যাচাই করতে। সেটা তো এমনিতে মেশিন করেনা, আমরা করি। ক্যালকুলেটর বারবার টেপার পরেও যখন কোনো সাড়া পাইনা, আমরা ক্যালকুলেটরের বাইরে, এমনকি বাড়ির বাইরে যাই, গিয়ে ব্যাটারি কিনে আনি, লাগাই, আবার চলে মেশিনটা। সব নিয়ন্ত্রণই আমাদের। অথচ ক্রমে কতদূর নিয়ন্ত্রণ করা যাচ্ছে দেখুন, পরশুদিন তথাগত ফোন করে জানাল ও একটা নতুন ইউপিএস কিনেছে, যেটা সরাসরি তার বিদ্যুৎ কতটা আছে সেই অবস্থাটা কারনেলকে জানিয়ে দিতে পারে। গ্নু-লিনাক্স কারনেল অপশনে দেখবেন এইসব দেওয়া থাকে। এবার ইউপিএস কখন চালু করা কখন বন্ধ করা দরকার সেটাও কারনেল দেখবে। তবে ভারি দাম। অথচ এই নিয়ন্ত্রণটা, আমাদের প্রয়োজনের দূরত্ব অব্দি, ব্যাশকে দিয়ে করানোই যেত। আসুন সেটা কী ভাবে করব তার ফ্লো চার্টটা বানানো যাক। ফ্লো-চার্ট কাকে বলে মনে আছে তো? কী ভাবে কাজটা করব তার ছক এবং গতি এবং ক্রিয়া কাঠামো। আমরা চাই এমন একটা প্রোগ্রাম যে নিজেই দেখে নেবে ফাইলটা আছে কিনা, কপি করা গেল কিনা। নইলে যেটা আমাদের নিজেদের করে নিতে হয়। কেন হলনা? ও দেখি তো ডিরেক্টরিটা, ফাইলটাই আছে কিনা। এবার সেটা ব্যাশ করবে। প্রোগ্রামটার কাঠামোটা হবে এইরকম —

```
if /etc/trash exists,
        copy it to ~/
        print "Done" to the Std. Out.
```

```
Otherwise
        print "file '/etc/trash' does not exist"
        exit
```

ঠিক আমরা মনুষ্যপদবাচ্য বলে পরিচিতরা যেমন যুক্তিকাঠামো বানাই। এবার এই কাঠামোটাকে ব্যাশে আনা যাবে যদি সিদ্ধান্ত নেওয়ার ওই নিয়ন্ত্রণ কাঠামোগুলো ব্যাশের স্ক্রিপ্টিং ল্যাংগুয়েজ দিয়ে নিয়ে আসা যায়। ঠিক এই কাজটাই করে লুপ। ব্যাশে সবচেয়ে প্রাথমিক এবং জরুরি রকমে যে যে লুপের নিয়ন্ত্রণকাঠামো পাওয়া যায় তারা হল, 'if', 'while', 'until', 'for', এবং 'case'। এই প্রতিটি লুপেরই একটা সুনির্দিষ্ট এবং সুস্পষ্ট, অঙ্কের ভাষায় বললে, ফাইনাইট এবং ডেফিনিট, শুরু এবং শেষ আছে, যা দিয়ে মেশিন জেনে যেতে পারে, কোথায় কোন কাজ শুরু হবে, এবং কোথায় কোনটা শেষ হবে।

আসুন প্রথমে এই ফ্লোচার্টটাকে একটা ব্যাশস্ক্রিপ্ট বানাই, যার একটা টেকনিকাল খুঁটিনাটি বুঝতে আপনাকে আরো দুতিনটে প্যারা ওয়েট করতে হবে। ওঃ ঈশ্বর, ওয়েটটা এখন আর এমনকি সেকশনের নয়, দিনের তো নয়ই, জাস্ট কয়েকটা প্যারার। এই স্ক্রিপ্টটা বানান 'trashcopy1' নামে একটা ফাইলে। তারপর তাকেও ঠিক আগের মতই করে চালনীয় করতে হবে এবং চালাতে হবে।

```
#!/bin/bash
if test -f /etc/trash
then
        #file exists, so copy and display
        cp /etc/trash ~/
        echo "Done"
else
        #file does not exist, so display error
        echo "There is no file '/etc/trash'"
        exit
fi
```

এই কাঠামোটা খেয়াল করুন। লুপটার গঠনটা হল 'if … then … else … fi'। 'if' দিয়ে শুরু, 'fi' দিয়ে শেষ। মধ্যে মধ্যে '#' চিহ্নটা দিয়ে যথারীতি মানবপাঠ্য এবং মানববোধ্য মন্তব্য আনা হয়েছে, ওই লাইনগুলো অব্যাশনীয়। মন্তব্যগুলো পড়ে দেখুন, সহজেই বুঝতে পারবেন, কী বলা এবং করা হয়েছে। শুধু একটা লাইন বুঝতে, বললামই তো, কয়েক প্যারা অপেক্ষা করতে হবে। এই স্ক্রিপ্টটা একদম কপিবুক স্ক্রিপ্ট, গাভাস্কারের ব্যাটিং-এর মত। সব নিয়ম মানা হয়েছে, সবসময় এত মানা হয়না। আর আগেই তো বলেছি, স্ক্রিপ্টের নামের এক্সটেনশন থাকা উচিত, '.sh'। আইন মেনে এই স্ক্রিপ্টটার নাম হওয়া উচিত ছিল, 'trashcopy1.sh'। আমার আর করা হয়না এসব। আসলে এগুলো তো কাজের স্ক্রিপ্ট না, জাস্ট আপনাদের স্ক্রিপ্ট পড়ার অভ্যেশ আনার জন্যে বানানো। আর আমার সব স্ক্রিপ্ট সবই থাকে হোম ডিরেক্টরির ভিতর '~/bin' বলে একটা ডিরেক্টরিতে। তাই বোঝা না-বোঝাটা এসব সমস্যা হয়না। কিন্তু ভালো না এসব ভালো না, বলেছেন শাক্যমুনি, বড়দাদের বাতেলা শুনতে শুনতেই বড় হওয়ার অভ্যেশ করাটাই হল ডিফল্ট। এবার চালান স্ক্রিপ্টটাকে। কী পেলেন?

```
There is no file '/etc/trash'
```

দেখছ কাকা, আমাদের স্ক্রিপ্টও বলতে শুরু করেছে, মহারাজ, আজ মনে হচ্ছে আমাদের বয়স হয়েছে। জিতা রহো বাচ্চা। স্ক্রিপ্টটার শরীরে ইনডেন্টগুলো দেখুন, এক একটা কাজের ব্লককে ইনডেন্ট করে ডানে সরিয়ে এক এক সঙ্গে আনা হয়েছে, এটাও স্ক্রিপ্টের কাজ করার নয়, স্ক্রিপ্ট পড়ে বোঝার সুবিধের স্বার্থে। একটা ব্লক 'then' অংশটার, আর একটা ব্লক 'else' অংশটার। কেন, বুঝতে পারছেন?

এবার ওই হিব্রু ভাষায় লেখা লাইনটা একটু মাপা যাক। যে মূল জিনিষটা ব্যবহার করা হয়েছে সেটা 'test', একটা প্রোগ্রাম। পাবেন ম্যানপেজের এক নম্বর সেকশনে। দেখুন ম্যানপেজের গৌরচন্দ্রিকাই বলে দিচ্ছে, টেস্ট কী করে — ফাইলের প্রকার পরখ করে এবং আলাদা আলাদা মানের ভিতর তুলনা করে, 'test - check file types and compare values'। এই ফাইল মানে যে কোনো রকম ফাইল, কোন ফাইল কী ধরনের সেটা যাচাই করে 'test'। এবং তুলনা করে একটা মানের সঙ্গে অন্য মানের, তারা সমান না বড় না ছোট না শূন্য এইসব দেখে। মান বলতে আঙ্কিক

ভ্যারিয়েবল বা স্ট্রিং ভ্যারিয়েবল মানে চিহ্ন-সমাহার দিয়ে তৈরি ভ্যারিয়েবলের মান। ম্যানপেজ থেকে ‘test’ প্রোগ্রামের অপশনগুলো পড়ে দেখুন, ‘-f’ অপশনটা যাচাই করে দেখে কোনো রেগুলার ফাইল আছে কিনা। এবার আমাদের দেওয়া নাম আর ধাম দেখে ব্যাশ ‘test’ কাজে লাগিয়ে যাচাই করে নিচ্ছে ওই ‘/etc’ ডিরেক্টরিতে ‘trash’ নামে কোনো ফাইল আছে কিনা। এই ‘trashcopy1’ ফাইলে আমরা কোনো ফাইল ওই নামে ওই ধামে আছে কিনা যাচাই করার জন্যে আদেশ দিয়েছিলাম, ‘test -f /etc/trash’। এই আদেশটা এর থেকে একটু অন্যরকম আকারেও দেওয়া যেত, ‘[-f /etc/trash]’। তখন স্ক্রিপ্টটার আকার দাঁড়াত

```
#!/bin/bash
if [ -f /etc/trash ]; then
        #file exists, so copy and display
        cp /etc/trash ~/
        echo "Done"
else
        #file does not exist, so display error
        echo "There is no file '/etc/trash'"
        exit
fi
```

চালিয়ে দেখুন, একই ভাবে চলবে। ‘test’ আদেশের এই আকারটা অনেক বহুলব্যবহৃত। যেমন বললাম, এই টেস্ট প্রোগ্রামটা ব্যবহার করা যায় দুটো ভ্যারিয়েবলের ভিতর তুলনায়, বা একটা ভ্যারিয়েবল এবং একটা কন্সটান্ট বা স্থির রাশির ভিতরে তুলনায়। যেমন ধরুন আমরা যদি আদেশ দিই ‘["$number -eq 5]’, টেস্ট তখন যাচাই করে দেখবে ‘number’ নামক ভ্যারিয়েবলের মান ‘5’-এর সমান কিনা। সত্যিই আমার কাজে লাগে এমন একটা স্ক্রিপ্ট তুলে দিই এখানে, যা এই ‘test’ ব্যাপারটা ব্যবহার করেছে। এখানে যেটা তুলে দিচ্ছি সেটা অবশ্য আমার আদত স্ক্রিপ্টটা নয়। তার সঙ্গে এমন কিছু যোগ করেছি যাতে বুঝতে সুবিধে হয় কী করে স্ক্রিপ্টটা কাজ করছে। কিছু মন্তব্য যোগ করেছি ওই ‘#’ চিহ্ন দিয়ে, সেটাও একই কারণে। আমার মেশিনে তিনটে গ্নু-লিনাক্স অপারেটিং সিস্টেমের জন্যে স্ক্রিপ্টটা লেখা, সুজে, স্ল্যাকওয়ার আর ম্যানড্রেক। কী নাম তাদের বা ঠিক তিনটেই কিনা তাতে অবশ্য কিছু এসে যায়না, দেখতেই পাবেন। এই প্রত্যেকটাতেই একজন করে ইউজার ‘dd’ আছে, মানে আমি। নানা সময়ে নানা কিছু করে দেখার জন্যে আমি আমার ব্যাশের লোকাল কনফিগারেশন ফাইলগুলোয় নানা বদল করি, কিন্তু সেই বদলগুলো মাঝে মাঝেই চূড়ান্ত ফ্লপ করে, তখন ম্যাও সামলানোর দরকার পড়ে। তাই এই কনফিগারেশন ফাইলগুলো, ব্যাকআপ করার দরকার পড়ে প্রতিটি সিস্টেমের ‘/home/dd’ ডিরেক্টরি থেকে। লোকাল কনফিগারেশন ফাইল মানে ‘~/.bashrc’ আর ‘~/.profile’। আমরা এখানে গোটা স্ক্রিপ্টটার পাশে বাঁদিকে পরপর বাংলায় লাইন নম্বরগুলো দিয়ে গেলাম, যাতে আলোচনা করার সুবিধে হয়।

০১	`#!/bin/bash`
০২	`rm /mnt/arkive/bkp/bash/bkp.*.*`
০৩	`#remove old backups`
০৪	`counter=1`
০৫	`#initialize the counter`
০৬	`for directory in / /mnt/slackware /mnt/mandrake`
০৭	`#first 'for' loop starts here`
০৮	`#for three systems in the machine, suse, slackware and mandrake,`
০৯	`#       mounted on /, /mnt/slackware and /mnt/mandrake`
১০	`do cd $directory`
১১	`#going into the three root directories, one by one`
১২	`echo "Now probing $directory"`
১৩	`# a check if the script is working properly`
১৪	`        if [ -d $directory/home/dd ];then`
১৫	`#first 'if' starts here`

```
১৬          echo "$directory/home/dd found"
১৭                  if [ -f $directory/home/dd/.bashrc ];then
১৮  #second 'if' starts here
১৯                  echo "'.bashrc' found in $directory/home/dd"
২০                  else echo "No '.bashrc' found in $directory/home/dd"
২১                  fi
২২  #second 'if ends here'
২৩                  if [ -f $directory/home/dd/.profile ];then
২৪  #third 'if' starts here
২৫                  echo "'.profile' found in $directory/home/dd"
২৬                  else echo "No '.profile' found in $directory/home/dd"
২৭                  fi
২৮  #third 'if' ends here
২৯          cd $directory/home/dd
৩০  #going into the '/home/dd' directory of the three systems, one by one
৩১                  for file in `ls .bashrc .profile`
৩২  #second 'for' starts here
৩৩                  do cp $file /mnt/arkive/bkp/bash/bkp$file.$counter
৩৪                  done
৩৫  #second 'for' ends here
৩৬          else echo "No $directory/home/dd found"
৩৭          fi
৩৮  #first 'if' ends here
৩৯  counter=$(expr $counter + 1)
৪০  #counter is incremented by 1 after every turn of first 'for' loop
৪১  echo "Counter is $counter"
৪২  #another kind of check
৪৩  done
৪৪  #first 'for' ends here
৪৫  cd /home/dd
৪৬  #back home real home
```

এবার দেখা যাক লাইন বাই লাইন। শুধু মূল স্ক্রিপ্টে যে লাইনগুলোয় মন্তব্য আছে সেগুলোকে বাদ দিয়েছি এই আলোচনা থেকে। যারা মূল স্ক্রিপ্টের মন্তব্যটুকু পড়েই কাজ করার রকমটা বুঝে যেতে পারছেন, তাদের আর এই লাইন ধরে ধরে আলোচনাটা পড়ার দরকারই নেই।

০১। এই লাইনটা সেই ধ্রুবপদ, ব্যাশস্ক্রিপ্টের বাধ্যতা।

০২। এই লাইনে ব্যাশকে বলা হচ্ছে, বাবা ব্যাশ, তুমি '/mnt/arkive/bkp/bash' ডিরেক্টরি থেকে কনফিগারেশন ফাইলের পুরোনো ব্যাকআপ ফাইলগুলো উড়িয়ে দাও।

০৪। গোটা স্ক্রিপ্ট জুড়ে আমরা যে কাউন্টারটা ব্যবহার করব, তাকে শূন্য করে নিচ্ছি, পুরো প্রক্রিয়ার ধাপগুলোকে যাতে গুনতে গুনতে যাওয়া যায়, ওই গোনা থেকে তৈরি সূচক আমরা পরে আমাদের ব্যাকআপ করা ফাইলগুলোর নামে ব্যবহার করব, সিস্টেম থেকে সিস্টেমে ফাইলগুলোকে আলাদা করার জন্যে। এই মুহূর্তে এই 'counter' ভ্যারিয়েবলের মান মানে '$counter' হল '1'। পরে ধাপে ধাপে বাড়বে, স্ক্রিপ্টের ৩৯ নম্বর লাইনে গিয়ে। প্রতিবার বাড়বে এক করে।

০৬। এই লাইনটা থেকে শুরু হচ্ছে প্রথম 'for' লুপের নিয়ন্ত্রণ কাঠামো। দেখুন এখানে তিনটে ডিরেক্টরির নাম দেওয়া আছে, '/', '/mnt/slackware', এবং '/mnt/mandrake'। মানে যেখানে যেখানে আমার মেশিনের তিনটে গ্নু-লিনাক্স সিস্টেমের তিনটে রুট পার্টিশন মাউন্ট করা আছে। এই ব্যাশ স্ক্রিপ্টটা বানানো সুজে সিস্টেমের ভিতরে দাঁড়িয়ে, যেখানে, '/mnt' ডিরেক্টরিতে দুটো সাবডিরেক্টরি '/mnt/slackware' আর '/mnt/mandrake' জুড়ে স্ল্যাকওয়ার আর ম্যানড্রেকের রুট পার্টিশন দুটো মাউন্ট করা আছে। এখানে দেওয়া ছকটা আমাদের গোটা বই জুড়ে ব্যবহৃত পুরোনো ছক থেকে একটু আলাদা, সেখানে কোনো ম্যানড্রেকই ছিলনা। এর মধ্যে আমার হার্ডডিস্কটা বদলে গেছে, মনে আছে? আর আগে তিনটে সিস্টেমের কথা আনতে চাইনি, জটিলতাটা বড্ড বেশি বেড়ে যাচ্ছিল বলে। এখন আমরা অনেকটা অভ্যস্ত হয়ে গেছি। এবং এই স্ক্রিপ্টটাকে আমার মেশিনের ব্যবস্থা থেকে একদম আলাদা ভাবেই পড়ুন। এমনকি সেই মেশিনে তিনটে ব্যবস্থা যদি নাও থাকে তাহলেও কিছু এসে যাবেনা, পরের গুলো সিস্টেম আর পাবেনা, কিন্তু গ্নু-লিনাক্স সিস্টেম থাকলে '/' থাকবেই, তার মানে অন্তত একবার স্ক্রিপ্টটা তার লুপে ঘুরে আসবেই। এবার এই 'for' লুপটার কাঠামো খেয়াল করুন। আমরা 'directory' বলে একটা ভ্যারিয়েবল তৈরি করছি, যার তিনটে আলাদা আলাদা মান হতে পারে '/', '/mnt/slackware', এবং '/mnt/mandrake'। এবার এই 'for' লুপটা ব্যাশকে বলছে তুমি পরপর এক একটা করে '$directory' মানে 'directory' ভ্যারিয়েবলের মান তোলো এবং সেই মান অনুযায়ী স্ক্রিপ্টের পরের লাইনগুলোয় দেওয়া আদেশগুলো চালাও। যেই একটা মানের জন্যে গোটা স্ক্রিপ্টের সবকটা আদেশ পালন করা শেষ হবে, তখন পরের মানটা নাও এবং আবার গোড়া থেকে শুরু করো। মনে পড়ছে আডার সেই কার্ডের কথা? কোনো মিল পাচ্ছেন?

১০। এই বার কাজ শুরু হচ্ছে। প্রথমে 'cd' করে সেই ডিরেক্টরিতে চলে যাও যার মান তোমার কাছে এই মুহূর্তে ভরা আছে 'directory' ভ্যারিয়েবলের জন্যে বরাদ্দ জমিতে। লুপটা যদি প্রথমবার চলছে তো এই মুহূর্তে '$directory' হল '/', তার মানে দাঁড়াচ্ছে — '`cd /`' আদেশটা পালন করো। তার মানে ব্যাশের ওয়ার্কিং ডিরেক্টরিটা বদলে গেল, খেয়াল করুন, এই মুহূর্তে সেটা '/'। লুপের দ্বিতীয় ঘোরায় এই জায়গায় এসে ব্যাশ যাবে '/mnt/slackware' ডিরেক্টরিতে, তৃতীয় ঘোরায় এসে যাবে '/mnt/mandrake' ডিরেক্টরিতে।

১২। এই লাইনটা মূল স্ক্রিপ্টে ছিলনা, আপনাদের জন্যে বানানো ভার্শনটায় ঢোকানো, অনেকটা মাছ ধরার ছিপের ফাতনার মত, তলায় তলায় ঘটনা কোন দিকে এগোচ্ছে তার উপরে নজরদারি করার একটা কৌশল। ভেবে দেখুন, লুপের প্রতি ঘোরায় একবার করে ব্যাশ এই লাইনে আসবে এবং ফুটিয়ে তুলবে সে কোন ডিরেক্টরিতে আছে, তিনবার তিনটে আলাদা নাম আসবে। স্ক্রিপ্টটা যে ঠিকঠাক এগোচ্ছে সেটা বোঝা যাবে।

১৪। এই প্রকৌশলটা তো আমরা শিখলাম এইমাত্র এই সেকশনেই। এটা বিচার করে দেখছে '$directory/home/dd' নামের কোনো ডিরেক্টরি আছে কিনা আদৌ। এই মুহূর্তে '$directory' হল '/', তার মানে আদতে আমরা '/home/dd' নামের ডিরেক্টরি খুঁজছি। লুপের পরের ঘোরাগুলোয় এই নামটা বদলে যাবে। এর ঠিক পরের বার নামটা হবে '/mnt/slackware/home/dd'। তার পরের বার কী হবে বলুন তো? এই লাইনেই শুরু হচ্ছে আর একটা নিয়ন্ত্রণ কাঠামো, 'if'। আমরা এই স্ক্রিপ্টে এই 'if' কাঠামোটা তিনবার ব্যবহার করেছি, তার প্রথমবার এই শুরু হল।

১৬। লাইন নম্বর ১৫-তে যদি ব্যাশ পরখ করে পায় যে আছে, আছে, অমন একটা ডিরেক্টরি আছে বৈকি, সঙ্গে সঙ্গে সেটা আমাদের জানিয়ে দেওয়ার ব্যবস্থা। নির্বাক ব্যাশকে একটু কথা বলতে শেখানো, যাতে তার পেটে পেটে কী ঘটছে সেটা আমরা জানতে পারি। এই লাইনটাও মূল স্ক্রিপ্টে ছিলনা, আপনারা চালাতে গেলে যাতে মূল কাঠামোটা ধরতে পারেন, তার জন্যে যোগ করা।

১৭। এই এলো দ্বিতীয় 'if'। খেয়াল করুন, প্রথম 'if' এখনো শেষ হয়নি, তার পেটের মধ্যে এই দ্বিতীয় 'if'। একে টেকনিকাল ভাষায় বলে নেস্টেড লুপ। লুপের বাচ্চা লুপ। তার মানে প্রথম 'if' যদি মেলে তবেই আমরা এই দ্বিতীয় 'if' নিয়ন্ত্রণে ঢুকছি। প্রথম 'if'-এর শর্তটাই যদি না মেলে তাহলে এই দ্বিতীয় 'if'-এ আমরা আদৌ ঢুকছি না। মানে '/home/dd' ডিরেক্টরি না-থাকলে দ্বিতীয় 'if' আমরা বিচার করছিই না, কারণ করার দরকারই পড়ছে না। দ্বিতীয় 'if' পরখ করে দেখছে '/home/dd' ডিরেক্টরিতে কোনো '.bashrc' ফাইল আছে কিনা। ভাবুন,

ডিরেক্টরিটাই না-থাকলে এটা পরখ করার দরকারই পড়ত না, সবাই তো আর চেশায়ার বিড়াল নয় যে আপনি না-থাকলেও আপনার হাসিটা রয়ে যাবে।

১৯। এই লাইনে জাস্ট ফাইলটা পাওয়া গেলে সেটা জানাতে বলা।

২০। যদি না পাওয়া যায়, তাও বলে দাও বাবা ব্যাশ।

২১। ইনি হলেন 'if'-এর শীর্ষাসন, মানে 'if' শেষ হওয়ার ইঙ্গিত। কোন 'if' শেষ হল এখানে? ঠিক এর আগেই যে 'if' শুরু হয়েছে, মানে দ্বিতীয় 'if'। 'fi' আমাদের জানাল, ১৭ নম্বর লাইনে বিসমিল্লা হওয়া এই স্ক্রিপ্টের দ্বিতীয় 'if' নিয়ন্ত্রণ কাঠামোটার ইন্তেকাল হল এই ২১ নম্বর লাইনে এসে।

২৩। এখান থেকে ২৭ নম্বর লাইন অব্দি হল এই স্ক্রিপ্টের তৃতীয় 'if' নিয়ন্ত্রণ। আর সব কিছু হুবহু এক দ্বিতীয় 'if'-এর সঙ্গে, শুধু এবার ব্যাশ খুঁজছে '.profile', এবং তার হ্যাঁ-সংবাদ বা না-সংবাদ জানিয়ে দিচ্ছে আমাদের। ২৭ নম্বর লাইনে এসে শেষ হচ্ছে এই তৃতীয় 'if'। প্রথম 'if' কিন্তু এখনো চলছে। মানে, তৃতীয় 'if'-ও একটা নেস্টেড ইফ। এবং এই প্রথম 'if' আবার চলছে প্রথম 'for' লুপের পেটের ভিতরে বসে। মনে আছে? প্রথম 'for' থেকেও আমরা এখনো বেরোইনি, যাতে ঢুকেছিলাম আমরা ০৬ নম্বর লাইনে।

২৯। এই লাইনে এসে ব্যাশ আবার তার ওয়ার্কিং ডিরেক্টরি বদলাচ্ছে। লাইন নম্বর ১০-এ ব্যাশ ঢুকেছিল '/' ডিরেক্টরিতে। তারপর '/home/dd' ডিরেক্টরির ফাইলের খবরাখবর নিচ্ছিল সেখানে বসেই। এবার ব্যাশ ঢুকছে '/home/dd' ডিরেক্টরিতে। এখন তার ওয়ার্কিং ডিরেক্টরি '/home/dd'। প্রথম 'for' লুপের দ্বিতীয় বার ঘোরায় এই লাইনে এসে ব্যাশ ঢুকবে '/mnt/slackware/home/dd' ডিরেক্টরিতে। শেষবার এই লাইনে এসে ব্যাশ কোন ডিরেক্টরিতে ঢুকবে বলুন তো?

৩১। এই আদেশ বদল বা কমান্ড সাবস্টিটিউশন আমাদের চেনা, '/home/dd' ডিরেক্টরিতে '.bashrc' বা '.profile' ফাইল যা খুঁজে পাবে ব্যাশ 'ls' করে, তাদের নিয়ে শুরু হচ্ছে এই দ্বিতীয় 'for' লুপ। এটাও নেস্টেড লুপ, কারণ প্রথম 'for' এখনো চলছে। চলছে প্রথম 'if' নিয়ন্ত্রণও। প্রথম 'for' লুপের ঘোরার সম্ভাব্য সর্বোচ্চ সংখ্যা তিন, এই দ্বিতীয় 'for' ঘোরার সম্ভাব্য সর্বোচ্চ সংখ্যা দুই, কারণ ফাইল থাকতে পারে দুটো। আর যদি একটা ফাইলও না-থাকে, তাহলে তো ল্যাটা চুকেই গেল। এই লুপ পয়দা হওয়া মাত্রই মরে গেল, ব্যাশ এগিয়ে যাবে স্ক্রিপ্টের পরবর্তী লাইনে। দেখুন, এই দ্বিতীয় 'for' লুপে ভ্যারিয়েবলের নাম হল 'file'। এই 'file' ভ্যারিয়েবলের তাই মান হতে পারে মাত্র দুটো, হয় '.bashrc' নয় '.profile'। ধরুন দুটো ফাইলই পেয়েছে ব্যাশ। তাহলে প্রথমবার ঘোরার সময় '$file' বা 'file' ভ্যারিয়েবলের মান কত? অবশ্যই '.bashrc'। দ্বিতীয়বার ঘোরার সময় '$file' দাঁড়াবে '.profile', যদি দুটো ফাইলই পাওয়া যায়।

৩৩। মজার কথা দেখুন, আসলে এই গোটা স্ক্রিপ্ট জুড়ে ব্যাশ যে কাজটা করবে সেটা আসলে এই লাইনটুকুই। যদি '.bashrc' আর '.profile' ফাইল দুটো পায়, বা একটাও পায় তাদের ভিতর থেকে, সেই ফাইল এবার ব্যাশ কপি করে দেবে '/mnt/arkive/bkp/bash' ডিরেক্টরিতে, যা অবশ্যই আগে থেকে বানানো আছে সিস্টেমে। আর ফাইল যদি না-পেয়ে থাকে সেটা তো ব্যাশ আগেই জানিয়ে দিয়েছে আমাদের। এবং কপি করার সময় ফাইলটার নাম বদলে দেবে। বদলের ছকটা হল 'bkp$file.$counter'। মানে নামটার আগে একটা 'bkp', পরে '.$counter'। এবার খেয়াল করুন, '$counter' এখন কত আছে। এটা প্রথম 'for' লুপের প্রথমবার ঘোরা চলছে, তার মানে '$counter' এখন '1'। আর '$file' এই মুহূর্তে '.bashrc'। তাহলে ব্যাক-আপ করা ফাইলের নাম দাঁড়াবে, তিনটে অংশ মিলিয়ে 'bkp.bashrc.1'। এই কপি করে চলা জুড়েই চলছে দ্বিতীয় 'for' লুপ।

৩৪। দ্বিতীয় 'for' লুপ শেষ হল, শেষ হওয়ার সিগনালটা আমরা চিনি, 'done'। প্রথম 'for' লুপ কিন্তু এখনো চলছে। এবং প্রথম 'if' নিয়ন্ত্রণও।

৩৬। এই আমরা এলাম আমাদের প্রথম 'if' নিয়ন্ত্রণের শেষ অংশে। ১৪ নম্বর লাইনে শুরু হওয়া 'if' নিয়ন্ত্রণের ভিতর আমরা এতটা সময় ধরে এগিয়েছি এই ধারায় যে '/home/dd' ডিরেক্টরিটা পাওয়া গেছে। কিন্তু যদি ডিরেক্টরিটা না পাওয়া যায় আদৌ? তাহলে কী হবে? এবার সেই দ্বিতীয় ধারা, ব্যাশ আমাদের পস্ট জানিয়ে দেবে, ওসব পাওয়া ফাওয়া যায়নি মামা, কপি কী করব, ডিরেক্টরিই নেই, এসব আওফাও কাজ দাও কেন?

৩৭। এসব বলার পরে ব্যাশের আর কী করার থাকে ‘fi’ ছাড়া? মানে টাটা, মানে প্রথম ‘if’ নিয়ন্ত্রণ শেষ হল এই লাইনে এসে।

৩৯। এই কলকব্জাটা তো আমরা ধরে ধরে আলোচনা করেছি আগেই, এক করে বাড়িয়ে দেওয়া ‘counter’ নামে ভ্যারিয়েবলটাকে। তার মানে খেয়াল করুন, এবার ‘$counter’ হয়ে গেল ‘2’। প্রথম ‘for’ লুপ পরের বার ঘোরায় আসব যখন আমরা তখন ব্যাক-আপ করা ফাইলে যোগ হবে ‘2’। ৩৩ নম্বর লাইনের সঙ্গে মেলান। তার মানে সুজের বেলায় হবে ‘1’। স্ল্যাকের বেলায় ‘2’। আর ম্যানড্রেকের বেলায় ‘3’। যদি না পাওয়া তাহলে তো সবচেয়ে ভালো, ব্যাক-আপ করতেই হবেনা।

৪১। এটাও একটা সেই দুধেভাতে-গার্ড। বোঝার সুবিধের জন্যে লাগানো।

৪৩। এতটা পথ পেরোলে লুপ খতম বলা যায়? প্রশ্নটা খুব সহজ এবং উত্তর-ও তো জানা। নিজেই করুন, নিজেই উত্তর দিন। কিন্তু একটা ঘোরা খতম মানেই পরের ঘোরার শুরু। প্রথম ‘for’ লুপ শুরু হয়েছিল ৬ নম্বর লাইনে। সেখানে ‘directory’ ভ্যারিয়েবলের যে তিনটে মান দেওয়া হয়েছিল, তার পরের মানটা নিয়ে শুরু হবে এবারের ঘোরা। এরকম চলবে, যতক্ষণ না সবকটা মান শেষ হয়।

৪৫। আপনি কোথায় বসে পড়ছেন? যদি বাড়িতে হয়, তাহলে বাইরে বেরিয়ে গিয়ে এই প্রথম গরমের রাস্তায় সিভিল ইঞ্জিনিয়ার আর আওয়ারা কুত্তাদের সঙ্গে মোলাকাত করে আসুন, নিজেই বুঝতে পারবেন এই লাইনটার মানে। (কার্টসি টেনিদা)।

তার মানে, ‘for … do … done’ লুপও শেষ হল, এই স্ক্রিপ্টটা বুঝতে গিয়ে। এবার দেখা যাক আর একটা ধরনের নিয়ন্ত্রণকাঠামো, ‘while ... do ... done’ লুপ। শুরু করা যাক একটা স্ক্রিপ্ট দিয়ে। এখন আমরা স্ক্রিপ্ট পড়ে অনেকটাই অভ্যস্ত হয়ে এসেছি। এই স্ক্রিপ্টটার আমি নাম দিয়েছিলাম ‘whileshow’। যে নামই দিন, সেটাকে চালনীয় বানান এবং চালান।

```
#!/bin/bash
while true; do
        echo "Press Ctrl-C to quit."
done
```

চালানো মাত্র কী দেখলেন, কোটি কোটি লাইন নেমে যাচ্ছে স্ক্রিনে, ‘`Press Ctrl-C to quit.`’। এবং সত্যিই এই প্রবাহ থেকে মুক্তি পাওয়ার একমাত্র উপায় ওই ‘<Ctrl><C>’ টাইপ করা। এখানে ব্যাশকে আমরা ব্যবহার করিয়েছি ‘test’-এর মতই আর একটা প্রোগ্রাম, তার নাম ‘true’। এই প্রোগ্রামটা খুব মজার, এই নিয়ে গ্নু-লিনাক্স জগতে অনেক চ্যাংড়ামিও আছে। ট্রু সবসময়েই ট্রু, যদিনা আপনি যে প্রক্রিয়াটাকে দিয়ে ‘true’ প্রোগ্রামটা ডেকে এনেছেন, সেই প্রক্রিয়াটাই বন্ধ করে দেন। এখানে আমরা যা করছি ‘<Ctrl><C>’ টাইপ করে। এই ‘true’-এরও ম্যানুয়াল পাবেন ম্যান পেজের সেকশন একে। পড়ে দেখুন। ট্রু কিছুই করেনা, কিন্তু এই না-করাটা অত্যন্ত সাফল্যের সঙ্গে করে — না না, এটা আমার নয়, ম্যানপেজের অভিমত। এই ‘while ... do ... done’ লুপটা দিয়ে আর একটা স্ক্রিপ্ট বানানো যাক, আমি নাম দিয়েছিলাম ‘whileshow1’।

```
#!/bin/bash
c=0
#initializing counter
while [ "$c" -le 10 ]; do
         echo "Now counter is: $c"
         c=$(expr $c + 1)
         # incrementing counter by one
         sleep 1
done
```

অনেকগুলো স্ক্রিপ্ট তো হল, এবার নিজে বোঝার চেষ্টা করুন তো, কী করছে এখানে? ধাপে ধাপে বাড়াচ্ছি আমরা কাউন্টার মানে ‘c’ ভ্যারিয়েবলটা, আর সেটাকে ফুটিয়ে তুলছি স্ক্রিনে, আর তারপর এক সেকেন্ড করে ব্যাশকে জিরোনোর ফুরশত দিচ্ছি প্রত্যেক ধাপেই। এর মধ্যে ‘test’ দিয়ে যে পরখ করে নেওয়া সেটাকে খেয়াল করুন, শর্তটা হল, ‘"$c" -le 10’। মানে কাউন্টার ভ্যারিয়েবলটার মান ‘-le 10’ মানে ‘10’-এর ছোট বা সমান কিনা, সেটাই দেখে

নিচ্ছে ব্যাশ। আগেই তো বলেছি 'test' কমান্ডের ম্যানুয়াল পড়ুন, সবগুলো খুঁটিনাটিই পেয়ে যাবেন। কাউন্টার ভ্যারিয়েবলটাকে আমরা জোড়াকোটের ভিতরে দিয়েছি, এটা বাধ্যতামূলক নয়, কিন্তু সুবিধাজনক, অনেক গোলযোগ কম হয়। এবার এই স্ক্রিপ্ট টাকে চালান। পাবেন নিচের এই লাইনগুলো। কিন্তু প্রতিটা লাইন আসবে পরের লাইনের এক সেকেন্ড পরে।

```
Now counter is: 0
Now counter is: 1
Now counter is: 2
Now counter is: 3
Now counter is: 4
Now counter is: 5
Now counter is: 6
Now counter is: 7
Now counter is: 8
Now counter is: 9
Now counter is: 10
```

এক সেকেন্ড পরে পরে লাইনগুলোর অবতীর্ণ হওয়াটা আমরা ঘটাচ্ছি ব্যাশকে ঘুম পাড়িয়ে পাড়িয়ে। কমান্ডটা হল 'sleep'। ম্যান পড়ার নেশা করো ম্যান, খেজুরগাছে হাড়ি বাঁধো ম্যান। তবে একটা ছুটকো চানস আছে গড়বড়ের। সব সিস্টেমে 'sleep' নাও পেতে পারেন, তখন খুঁজুন 'usleep'। স্ক্রিপ্টটা সেটা দিয়েও বানিয়ে নিতে পারবেন। এইমাত্র করা প্রোগ্রামটাই অন্য একটা লুপ দিয়েও করা যেত, 'until ... do ... done'।

```
#!/bin/bash
c=0
until [ "$c" -ge 10 ];do
        echo "Now counter is: $c"
        c=$(expr $c + 1)
        sleep 1
done
```

এখানে খেয়াল করুন, আঙ্কিক যুক্তিটা একটু উল্টে দেওয়া হয়েছে, 'test' এখন পরখ করছে, '"$c" -ge 10' কিনা, মানে 'c' ভ্যারিয়েবলের মান দশের চেয়ে বড় কিনা? এটা কী হল? ভাবুন, লুপটাকে, হোয়াইল হল 'যখন', আর আনটিল হল 'যতক্ষণ না'। যখন আমি দশের ছোট, তার মানেই তা যতক্ষণ না আমি দশের বড় বা সমান।

এখন আমাদের বাকি আর একটাই লুপের কথা, সেটা হল 'case … in … done'। এর একটা উদাহরণ দিয়ে এসেছি আগেই, 'xpick' নামে, নয় নম্বর দিনে।

```
#!/bin/bash
echo "Choose a number to pick your Window Manager"
echo ""
echo "1.  KDE"
echo ""
echo "2. GNOME"
echo ""
echo "3. FLUXBOX"
echo ""
echo "4. BLACKBOX"
echo ""
read NUMBER
case $NUMBER in
   1) export WINDOWMANAGER=startkde ;;
   2) export WINDOWMANAGER=gnome-session ;;
   3) export WINDOWMANAGER=fluxbox ;;
   4) export WINDOWMANAGER=blackbox ;;
   *) echo ""; echo "Are You Literate? Try Again." ; echo "" ; exit ;;
esac
exec startx
```

চালান স্ক্রিপ্টটা। দেখুন, ব্যাশ স্ক্রিনে একটা মেনু তৈরি করে দিচ্ছে। মধ্যে এক এক লাইন ফাঁকা জায়গা ছেড়ে নিচের এই চারটে লাইন, পরপর।

```
1. KDE
2. GNOME
3. FLUXBOX
4. BLACKBOX
```

এবার এই মেনু মিলিয়ে যে নম্বর আপনি টিপবেন সেই রকমের এক্স-উইনডোজ চালু হবে। এর মধ্যে এক্স-উইনডোজ সংক্রান্ত যে জটিলতাটা আছে সেটা ছেড়ে দিন, অন্যটুকুকে ভাবুন। আপনি '`NUMBER`' বলে একটা ভ্যারিয়েবল তৈরি করলেন, তার চার চারটে আলাদা মান হতে পারে। '`$NUMBER`' হতে পারে '`1`' বা '`2`' বা '`3`' বা '`4`'। এর বাইরেও হতে পারে, সম্পূর্ণ অন্য কিছু, তখন আপনি তাকে মাইল্ড গালি দিয়ে নিচ্ছেন, '`Are You Literate? Try Again.`'। এবং এই পাঁচ নম্বর সম্ভাবনাটা ঘটলে স্ক্রিপ্টটা এখানেই শেষ হয়ে যাচ্ছে। আপনাকে আবার '`./xpick`' বলে নতুন করে শুরু করতে হচ্ছে। মানে এই লুপ যাকে '`case`' বলে ডাকছে তার এই পাঁচটা সম্ভাবনা। সম্ভাবনাগুলোকে মিলিয়ে লুপটা শুরু '`case $NUMBER in`' লাইন থেকে এবং শেষ '`esac`' লাইনে। ঠিক '`case`' কথাটার উল্টো। ইফ লুপে যেমন দেখেছি আমরা। 'if' লুপের সঙ্গে খুবই মিল 'case' লুপের। এই প্রোগ্রামটা 'if' লুপ দিয়েও করা যেত। 'NUMBER' ভ্যারিয়েবলের প্রতিটি মানের জন্যে একটা করে 'if' শর্ত লিখতে হত, মানে লেখার খাটনিটা একটু বেশি হত।
এবার আমরা তৈরি, আপনার সিস্টেমেই যে স্ক্রিপ্টগুলো দেওয়া আছে, অজস্র, সেগুলো পড়ার কাজে হাত দেওয়ায়। মানে, আপনি এখন আপনার সিস্টেম পড়া এবং শেখা শুরু করতে পারেন, মানে আমার কাজ খতম। পয়লা নভেম্বর থেকে ধরলে, আজ বাইশে মার্চ, মানে একশো বাহাত্তর দিনের আমাদের এই ম্যারাথন খতম। যদিও, যেমন বলেছিলাম, আপনাকে একটু গালাগাল দেওয়ার ব্যাপার আছে, সেটা আসছে ছোট হরফে, প্রতি দিনের শেষেই যেমন এসেছে।

প্রথমে বলি একটা ভালো কথা। শুধু বইটার জন্যে না, আমার জন্যেও ভালো। আজকাল ভালো কিছু এত কম দেখি, একটু কিছু দেখলেই মনটা ভারি ভরে যায়। এই বইটা লিখে চলাটা আমায় একটা অন্য অনুভব দিয়েছে। আত্মীয়তার অন্য একটা মাত্রা। বইটার জন্যে আমার কী পরিমাণ খাটনি গেছে সেটা আপনারা আন্দাজ করতে পারছেন। সেই খাটনিটা বোধহয় দিতেই পারতাম না এই অভিজ্ঞতাটাকে বাদ দিয়ে। লেখা এবং পড়া — গত বেশ কিছু বছর মোটামুটি ভাবে এটাই আমার কাজ। বই লেখার প্রক্রিয়াটা সেখানে খুবই রোজকার একটা অভিজ্ঞতা, নিজে না হোক অন্যে, আমার চারপাশে নিয়তই লেখা হচ্ছে, এই বই বা ওই বই। কিন্তু এতগুলো বছরের সেই বইঅভিজ্ঞতাতে এরকম কিছু আমি কখনো দেখিনি। কারুর না। না আমার বেলায়, না অন্য-কারুর বেলায়। সেটা এই সামগ্রিকতার অনুভব।

ধরুন এই বইটা লিখেছি শেষ অব্দি আমি, আমার মেশিনে, আমার কিবোর্ডে। আমি যদি বইটার অনেকটা মা হই, বেশ কিছুটা মা সঙ্কর্ষণ, কিছুটা করে মা তথাগত অশেষ এরা। এটা কী করে সম্ভব তাও আমি ভালো বুঝিনি। প্রতি দিন, প্রতিটা দিন, যে প্রক্রিয়ায় সঙ্কর্ষণরা বইটা বিয়োনোর কাজে পার্ট নিয়েছে, সেটা আমি কোনোদিন কারোর বইয়ের বেলাতেই করিনি, করার কথা ভাবিওনা, আসলে আমি ওদের চেয়ে অনেকটা বুড়ো হয়ে গেছি, তাই অনেকটা নীচও।

এবার, যে গালাগালটা আপনাদের দেওয়ার, তার সঙ্গে এর একটা খুব নিকট সম্পর্ক। এই যত্নের প্রক্রিয়ার মধ্যেই সঙ্কর্ষণরা সেই সবই করেছে এই বইয়ের জন্যে যা যা করার কথা, এবং যা যা করার কথা নয়, বোধহয় তাও। সেরকম চেষ্টার ঠেলাতেই হয়ত, একটা সম্ভাবনা দেখা দিয়েছে, ফ্রি সফটওয়ার ফাউন্ডেশন থেকে এই বইটা প্রকাশের, হয়ত, এখনো নিশ্চিত কিছু না, এবং হয়ত, বাংলার পাশাপাশি ইংরিজি মালয়ালম ইত্যাদি অনুবাদেও। কিন্তু বেদনাটা ঠিক এইখানেই। আজ যদি এর অন্য ভাষায় অনুবাদ হয়, ইংরিজি হলে তো বটেই, মলয়ালম হিন্দি তামিল ইত্যাদির বেলাতেও তাই, একটা অপমান তাদের মেনে নিতে হবেনা, এই বইটা লেখার একদম গোড়া থেকেই যেটা ছিল। আজ যদি এই বইটা আমি গোড়া থেকেই ইংরিজিতে লিখতাম তাহলেও যে অপমানটা হতনা।

বইটা লেখা হচ্ছে গ্নু-লিনাক্স নিয়ে, গ্নু-লিনাক্সের অতুলনীয় শক্তিমানতায় অভিভূত হয়ে, অথচ গ্নু-লিনাক্সে কিন্তু লেখা যাচ্ছেনা। অঙ্কুর বাংলা নিয়ে সায়মিন্দু-সঙ্কর্ষণদের প্রচুরতর পরিশ্রমের পরেও, আজো গ্নু-লিনাক্সে, ওপনসোর্স জগতে বাংলায় খুব ভালো কিছু উপায় নেই। যা আছে তাতে এত বড় একটা লেখা শুধু অতুলনীয় রকমের কষ্টসাধ্য নয়, বেশ কিছুটা বোকামিও। এত বেশি নিজেকে অপব্যয় করে চলতে হবে সেইসব উদ্ভট প্রক্রিয়ায়। অথচ, ইংরিজি তো ছেড়েই দিন, তামিল মলয়ালম হিন্দিতে কিন্তু আছে। কেন? আপনার জন্যে। আপনিও আছেন এই বইটায়, আপনার অনড় এবং সংবেদনহীন অনুপস্থিতি নিয়ে, যদি আপনি থাকতেন, আপনি, আপনি, আপনারা প্রত্যেকেই, তাহলে এই বইটা বাংলাতেই লেখা যেত, গ্নু-লিনাক্সেই। ভারতের বেশ কিছু আঞ্চলিক ভাষাতেই গ্নু-লিনাক্সে কাজ করার উপায় যে তৈরি হয়েছে, সেটা আকাশ থেকে পড়েনি। আমার সত্যিই , শুধু খারাপ লাগা নয়, বেশ লজ্জাও লাগছে, আপনার কারণে।

glt-mad@ilug-cal.org

MADHYAMGRAM

সংকলন ও রচনা : মধ্যমগ্রাম জিএলটি-র (glt-mad@ilug-cal.org) তরফে : ত্রিদিব সেনগুপ্ত